



NEURAL EXTENSIONS TO ROBUST PARAMETER DESIGN

DISSERTATION

Bernard Jacob Loeffelholz

AFIT/DS/ENS/10-03

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

The views expressed in this dissertation are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

AFIT/DS/ENS/10-03

NEURAL EXTENSIONS TO ROBUST PARAMETER DESIGN

DISSERTATION

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the
Degree of Doctor of Philosophy in Operations Research

Bernard Jacob Loeffelholz, B.S., B.A., M.S.

September 2010

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

NEURAL EXTENSIONS TO ROBUST PARAMETER DESIGN

Bernard Jacob Loeffelholz, B.S., B.A., M.S.

Approved:

_____/signed/_____ Dr. Kenneth W. Bauer, Jr. Committee Chairman	____ 30 Jul 2010 ____ Date
---	-------------------------------

_____/signed/_____ Dr. John O. Miller Committee Member	____ 30 Jul 2010 ____ Date
--	-------------------------------

_____/signed/_____ Dr. Mark Oxley Committee Member	____ 30 Jul 2010 ____ Date
--	-------------------------------

Accepted:

_____/signed/_____ M. U. Thomas Dean, Graduate School of Engineering and Management	____ 16 Aug 2010 ____ Date
--	-------------------------------

Abstract

Robust parameter design (RPD) is implemented in systems in which a user wants to minimize the variance of a system response caused by uncontrollable factors while obtaining a consistent and reliable system response over time. Typically, quadratic regression is deemed sufficient to specify a process model of model system behavior. We propose the use of artificial neural networks (ANNs) to compensate for highly non-linear problems that quadratic regression fails to accurately model.

RPD is conducted under the assumption that the relationship between the system response and controllable and uncontrollable variables does not change over time. Since degradation in the system response will almost certainly occur; this assumption will inevitably be violated. We propose a methodology to find a new set of settings that will be robust to moderate system degradation while remaining robust to noise variables within the system. An algorithm is presented for this enhanced RPD analysis utilizing both quadratic regression and two specific artificial neural network architectures.

RPD has been well developed on single response problems. Sparse literature exists on dealing with multiple responses in RPD and most methods utilize a subjective weighting scheme. To account for multiple responses, we examine the use of factor analysis on the response data. Linear combination techniques are also developed in the case that more than a single factor is retained in the analysis.

All the proposed techniques are applied to textbook applications to demonstrate their utility. An Air Force application problem is then examined to demonstrate the new

technique's potential on a real-world problem that is highly non-linear. The application is a detector developed to detect anomalies within hyper-spectral imagery.

The results of this research include successful implementation of artificial neural networks in RPD. These artificial neural networks can be utilized when faced with a highly non-linear problem. Also, new settings are developed that are shown to be superior to traditional robust settings when a system is subject to performance degradation. A new methodology of approaching multiple response problems is developed which shows promise. Finally, the anomaly detector is further enhanced through the use of artificial neural networks to determine robust settings and alternate settings when degradation is expected.

Acknowledgments

I would like to thank Dr. Kenneth Bauer for his extraordinary vision, expertise, guidance, and most of all, his patience during this process. I am grateful for him allowing me to utilize him while undertaking this great educational milestone in my life. I would also like to thank Dr. J.O. Miller and Dr. Mark Oxley for the time and dedication they put into my dissertation.

A tremendous amount of thanks goes to my fellow classmate, Capt. Earl Bednar. Without his friendship and assistance, this process would not have been possible. I have thoroughly enjoyed our time in courses together and the many hours we have spent talking about our research endeavors. I also have a special thanks to Maj. Frank Mindrup for his friendship and professional collaboration, especially near the end of this research. Without his dedication and patience, finishing this research would have seemed out of reach. The many other classmates have also been instrumental and I have enjoyed their friendships the last few years.

I would also like to thank my parents and the rest of my family. Without their continuous support, I would not be where I am today. They have truly laid the foundation for me to succeed.

Most importantly I would like to thank my wife and my boys. Without them, I would not be the person I am today. My wife has sacrificed a lot for me and has continuously supported me throughout this process. I love her with all my heart and am so thankful she is in my life.

Jacob Loeffelholz

Table of Contents

	Page
Abstract.....	iv
Acknowledgments.....	vi
List of Figures	ix
List of Tables	xi
List of Abbreviated Terms	xiv
 I. Introduction	 15
1.1 General Discussion	15
1.2 Motivation	16
1.3 Research Goals	16
1.4 Proposed Research Contributions.....	17
1.5 Organization of Dissertation.....	18
 II. Literature Review	 20
2.1 Overview	20
2.2 Robust Parameter Design	20
2.2.1 <i>Crossed Array Design</i>	22
2.2.2 <i>Combined Array Design</i>	26
2.2.3 <i>Response Surface Methodology</i>	28
2.2.4 <i>Solving the Dual Response Problem</i>	31
2.2.5 <i>Robustness</i>	37
2.3 Artificial Neural Networks	38
2.3.1 <i>ANN Classification and Regression</i>	38
2.3.2 <i>Radial Basis Function Neural Networks</i>	39
2.3.3 <i>Generalized Regression Neural Networks</i>	43
2.4 Multiple Responses.....	45
2.4.1 <i>Weighting the Responses</i>	46
2.4.2 <i>Visualizing the Responses</i>	47
2.4.3 <i>Desirability Functions</i>	47
2.4.4 <i>MSE and PCA</i>	49
2.5 Factor Analysis	50
2.5.1 <i>Factor Analysis and PCA</i>	51
2.5.2 <i>Mathematical Model of Factor Analysis</i>	51
2.5.3 <i>Extracting Factors</i>	52
2.5.4 <i>Factor Rotation and Factor Scores</i>	52
2.5.5 <i>Summary</i>	53
 III. Methodology	 54
3.1 Overview	54
3.2 Robust Parameter Design	54
3.2.1 <i>RPD Example Problem</i>	56

3.3 Robust Parameter Design and System Degradation	61
3.3.1 <i>Guarding Against System Degradation</i>	61
3.3.2 <i>Doubly Robust Solution Example</i>	69
3.3.4 <i>RPD Summary</i>	76
3.4 Artificial Neural Networks	77
3.4.1 <i>ANNs and RPD</i>	78
3.4.2 <i>Semiconductor Extended Example Using ANNs in RPD</i>	83
3.4.3 <i>Koksoy Problem Using ANNs in RPD</i>	89
3.4.4 <i>Doubly Robust Operating Points Using ANNs</i>	91
3.5 Multiple Responses.....	101
3.5.1 <i>Factor Analysis for Multiple Responses</i>	102
3.5.2 <i>Factor Analysis Problem</i>	106
IV. Application	113
4.1 Overview	113
4.2 Hyper-spectral Imagery	113
4.3 AutoGAD.....	118
4.3.1 <i>AutoGAD Outputs</i>	119
4.3.2 <i>AutoGAD Control Variables</i>	120
4.3.3 <i>AutoGAD Noise Variables</i>	121
4.3.4 <i>AutoGAD Setup Summary</i>	122
4.4 AutoGAD and RPD	123
4.4.1 <i>AutoGAD RPD Design</i>	124
4.4.2 <i>AutoGAD Quadratic Regression RPD</i>	125
4.4.3 <i>AutoGAD ANN RPD</i>	134
4.5 AutoGAD and System Degradation	141
4.5.1 <i>Doubly Robust Settings in Quadratic Regression</i>	142
4.5.2 <i>Doubly Robust Settings in ANNs</i>	145
4.6 AutoGAD and Factor Analysis	149
V. Contributions and Future Research.....	160
5.1 Overview	160
5.2 Research Contributions.....	160
5.2.1 <i>Doubly Robust Settings</i>	160
5.2.2 <i>Artificial Neural Networks in RPD</i>	161
5.2.3 <i>Factor Analysis in RPD</i>	161
5.3 Recommendations for Future Research.....	161
5.3.1 <i>Robust Parameter Design</i>	161
5.3.2 <i>Artificial Neural Networks</i>	162
5.3.3 <i>Multiple Responses in RPD</i>	162
Appendix A. Summary of Example Problems.....	163
Appendix B. CCD for AutoGAD.....	164
Bibliography	165

List of Figures

	Page
FIGURE 1. CONTROL AND NOISE VARIABLE INTERACTION PLOT	22
FIGURE 2. CROSSED ARRAY DESIGN (MYERS & MONTGOMERY, 2002).....	23
FIGURE 3. CENTRAL COMPOSITE DESIGN WITH AXIAL POINTS = 1	27
FIGURE 4. RBFNN WITH SINGLE OUTPUT	40
FIGURE 5. GENERALIZED REGRESSION NEURAL NETWORK	45
FIGURE 6. MEAN CONTOUR PLOT	59
FIGURE 7. VARIANCE CONTOUR PLOT	59
FIGURE 8. OVERLAY CONTOUR PLOT	60
FIGURE 9. GRADIENT SEARCH FOR NEW SOLUTION	64
FIGURE 10. ALGORITHM FOR FINDING A DOUBLY ROBUST OPERATING POINT	68
FIGURE 11. LT CONTOUR PLOT FOR SEMICONDUCTOR EXAMPLE PROBLEM	73
FIGURE 12. LT CONTOURS FOR ORIGINAL LT SETTINGS	74
FIGURE 13. LT CONTOURS FOR DOUBLY ROBUST SETTINGS	74
FIGURE 14. $LT(x^{**}) - LT(x^*)$ CONTOUR PLOT.....	76
FIGURE 15. APPROACH 1 TO DEVELOP ANN FOR RPD	80
FIGURE 16. APPROACH 2 TO DEVELOP ANN FOR RPD	82
FIGURE 17. SCATTERPLOT OF OPTIMAL LT SETTINGS OVER 100 REPLICATIONS	84
FIGURE 18. NORMAL PROBABILITY PLOT FOR SEMICONDUCTOR EXTENDED EXAMPLE	87
FIGURE 19. PREDICTED VS ACTUAL FOR SEMICONDUCTOR EXTENDED EXAMPLE	88
FIGURE 20. LT ANN FOR GRADIENT ANALYSIS	93

FIGURE 21. RBFNN WITH SINGLE OUTPUT (LT)	96
FIGURE 22. GRNN WITH SINGLE OUTPUT (LT)	97
FIGURE 23. ALGORITHM FOR SYSTEM DEGRADATION IN ANNs	99
FIGURE 24. ELECTROMAGNETIC SPECTRUM (PABICH, 2002)	114
FIGURE 25. HSI RANGE IN EM SPECTRUM (LANDGREBE, 2003).....	115
FIGURE 26. HSI CUBE EXAMPLE (SHAW ET AL., 2002)	116
FIGURE 27. LAYERS OF DATA IN SPECTRAL DIMENSION (MILLER, 2009)	116
FIGURE 28. TRANSFORMING HSI CUBE INTO DATA (MILLER, 2009)	117
FIGURE 29. AUTOGAD ALGORITHM.....	118
FIGURE 30. RESIDUAL PLOTS FOR TIME, TPF, FPF, TFP	133
FIGURE 31. SPREAD VS. RMSE FOR TPF	135

List of Tables

	Page
TABLE 1. CURRENT METHODOLOGIES TO SOLVE DUAL RESPONSE PROBLEM IN RPD.....	36
TABLE 2. TYPES OF FACTOR EXTRACTION METHODS	52
TABLE 3. SEMICONDUCTOR EXAMPLE PROBLEM	57
TABLE 4. LT SETTINGS AND SOLUTION TO SEMICONDUCTOR PROBLEM.....	58
TABLE 5. M&M SOLUTION AND LT SOLUTION TO SEMICONDUCTOR PROBLEM.....	58
TABLE 6. ORIGINAL AND DOUBLY ROBUST SETTINGS	72
TABLE 7. NOTIONAL DESIGN MATRIX	78
TABLE 8. PRE-PROCESSED DATA FOR MEAN AND VARIANCE.....	81
TABLE 9. TRUE OPTIMAL SETTINGS FOR EQUATION (3.27).....	84
TABLE 10. ANOVA FOR SEMICONDUCTOR EXTENDED EXAMPLE.....	85
TABLE 11. COEFFICIENTS FOR SEMICONDUCTOR EXTENDED EXAMPLE.....	86
TABLE 12. ANN RESULTS ON SEMICONDUCTOR EXTENDED EXAMPLE	89
TABLE 13. EXAMPLE PROBLEM 2 (KOKSOY, 2008).....	90
TABLE 14. ESTIMATED OPTIMAL SETTINGS FOR KOKSOY EXAMPLE	90
TABLE 15. SETTINGS FOR ANN AND QR FOR EXAMPLE 2	91
TABLE 16. ANN DOUBLY ROBUST SETTINGS UNDER NORMAL OPERATING CONDITIONS	101
TABLE 17. SUMMARY OF (ROTATED) FACTOR SCORE REDUCTIONS	105
TABLE 18. OPTIMAL SETTINGS FOR FIVE RESPONSE PROBLEMS	107
TABLE 19. SETTINGS FROM QR AND ANN FOR FIVE RESPONSES	107
TABLE 20. FACTOR LOADINGS MATRIX.....	108

TABLE 21. ROTATED FACTORS LOADINGS MATRIX.....	108
TABLE 22. ROBUST SOLUTIONS FOR (ROTATED) FACTOR REDUCED METHODS.....	109
TABLE 23. LT VALUES FOR FACTOR ANALYSIS PROBLEM	110
TABLE 24. PERCENTAGE FROM OPTIMAL LT VALUES.....	111
TABLE 25. CONTROL VARIABLES IN AUTOGAD.....	121
TABLE 26. OPTIMAL SETTINGS FOR 4 RESPONSES SUGGESTED BY QR	128
TABLE 27. TESTING RESULTS FOR OPTIMAL SETTINGS IN QR	129
TABLE 28. ANOVA FOR TIME.....	131
TABLE 29. ANOVA FOR TPF	131
TABLE 30. ANOVA FOR FPF	132
TABLE 31. ANOVA FOR TFP	132
TABLE 32. OPTIMAL SETTINGS FOR FOUR RESPONSES (RBFNN).....	137
TABLE 33. OPTIMAL SETTINGS FOR FOUR RESPONSES (GRNN).....	137
TABLE 34. TESTING RESULTS FOR OPTIMAL SETTINGS USING RBFNN	138
TABLE 35. TESTING RESULTS FOR OPTIMAL SETTINGS USING GRNN	138
TABLE 36. OPTIMAL SETTINGS FOR QR AND ANNs	138
TABLE 37. TESTING RESULTS FOR QR AND ANNs ON TIME.....	139
TABLE 38. TESTING RESULTS FOR QR AND ANNs ON TPF	139
TABLE 39. TESTING RESULTS FOR QR AND ANNs ON FPF	139
TABLE 40. TESTING RESULTS FOR QR AND ANNs ON TFP	139
TABLE 41. ORIGINAL AND DOUBLY ROBUST SETTINGS FOR TPF	143
TABLE 42. IMAGE 6 RESULTS FOR TPF SETTINGS.....	143
TABLE 43. ORIGINAL AND DOUBLY ROBUST SETTINGS FOR TFP	144

TABLE 44. IMAGE 6 RESULTS FOR TFP SETTINGS	144
TABLE 45. RBFNN ORIGINAL AND DOUBLY ROBUST SETTINGS FOR TPF	146
TABLE 46. RBFNN IMAGE 6 RESULTS ON TPF.....	146
TABLE 47. GRNN ORIGINAL AND DOUBLY ROBUST SETTINGS ON TPF	147
TABLE 48. GRNN IMAGE 6 RESULTS ON TPF.....	147
TABLE 49. RBFNN ORIGINAL AND DOUBLY ROBUST SETTINGS ON TPF	148
TABLE 50. RBFNN IMAGE 6 RESULTS ON TPF.....	148
TABLE 51. GRNN ORIGINAL AND DOUBLY ROBUST SETTINGS ON TPF	149
TABLE 52. GRNN IMAGE 6 RESULTS ON TPF.....	149
TABLE 53. EIGENVALUES OF FACTORS FOR AUTOGAD.....	151
TABLE 54. FACTORS LOADINGS MATRIX FOR 2 FACTORS	152
TABLE 55. ROTATED FACTORS LOADINGS MATRIX FOR 2 FACTORS.....	152
TABLE 56. FACTORS LOADINGS MATRIX FOR 3 FACTORS	153
TABLE 57. ROTATED FACTORS LOADINGS MATRIX FOR 3 FACTORS.....	153
TABLE 58. SUMMARY OF LINEAR COMBINATIONS ON FACTOR SCORES	154
TABLE 59. LT SETTINGS FOR COMBINED RESPONSE DATA.....	155
TABLE 60. LT SETTINGS FOR TWO FACTORS USING QUADRATIC REGRESSION.....	155
TABLE 61. LT SETTINGS FOR TWO FACTORS USING ANNs	156
TABLE 62. LT RESULTS FOR TWO FACTORS	157
TABLE 63. LT SETTINGS FOR THREE FACTORS USING QUADRATIC REGRESSION.....	158
TABLE 64. LT SETTINGS FOR THREE FACTORS USING ANNs	158
TABLE 65. LT RESULTS FOR THREE FACTORS	159

List of Abbreviated Terms

ANN – Artificial Neural Network

BPN – Back Propagation Network

CCD – Central Composite Design

DOE – Design of Experiments

FA – Factor Analysis

FFNN – Feed Forward Neural Network

GRNN – Generalized Regression Neural Network

LT – Lin & Tu Methodology

MSE – Mean Square Error

PCA – Principal Component Analysis

QR – Quadratic Regression

RBFNN – Radial Basis Function Neural Network

RPD – Robust Parameter Design

RSM – Response Surface Methodology

SNR – Signal to Noise Ratio

NEURAL EXTENSIONS TO ROBUST PARAMETER DESIGN

I. Introduction

1.1 General Discussion

Robust parameter design (RPD) determines a set of control variable settings that minimize the variance of the response caused by different sources of noise in a system while satisfying the constraint on the mean (Myers & Montgomery, 2002). The idea is that second order models capture the mean and variance of the system response and that these models do not change in time.

Many Air Force applications involve modeling systems with a large number of control settings outputting multiple responses. One such application was created by Johnson (2008) which is an autonomous global anomaly detector (AutoGAD). In its current version, the detection algorithm implemented in MATLAB® contains eleven control variables and four responses. Davis (2009) first applied RPD on AutoGAD to determine robust control settings and promising results were realized. However, ANOVA analysis suggested the use of quadratic regression was inadequate to predict true response values.

In AutoGAD, RPD assumes that new information will closely resemble the training data. This may not be an appropriate assumption in many applications. For example, the detection algorithm could encounter an image “noisier” than any image in its library. In the case of AutoGAD, such an occurrence corresponds to “system degradation.”

The main objective of this research is to determine a new set of control variable settings that are not only robust to noise in the system, but also are robust to a moderate amount of system degradation. In mechanical systems, this degradation could be of a physical nature. In software systems, the degradation is represented as being exposed to inputs beyond their experience and training. Other objectives of this research suggest the use of artificial neural networks as an alternative to quadratic regression to model the process. Finally, the problem of multiple responses is explored.

1.2 Motivation

The motivation behind this research is to apply appropriate robust settings to the detection algorithm to improve its performance. The detection algorithm is currently employed based on the settings suggested by Johnson (2008). These settings were based on the experience of the author and tended to maximize only one of the four available responses in detection. Further, the settings were based on only eight given images, thus the author used the same images in the testing set as the training set. This situation typically leads to an overly optimistic view of system performance.

1.3 Research Goals

The first goal of this research was to determine a new set of “doubly robust” settings that are robust to noise variables and system degradation. The algorithm constructed in this research was applied to the detection algorithm, AutoGAD, created by Johnson (2008). Comparisons were made from these doubly robust settings to traditional RPD settings in the presence of system degradation.

A secondary goal of this research was to examine the use of artificial neural networks (ANNs) as a substitute for quadratic regression in RPD. These ANNs were compared with quadratic regression to determine their ability to better fit highly non-linear models. Robust and doubly robust settings were also calculated.

A third goal of this research was to determine a new method of combining multiple responses into a single dimension. Factor analysis was explored as the appropriate technique in determining commonalities among various responses. Finally, if more than one factor was retained, linear combination methods were suggested to combine multiple factor scores into a single dimension.

1.4 Proposed Research Contributions

An algorithm to determine new “doubly robust” settings that can be applied to problems containing control and noise variables was developed in this research. To solve the dual response problem of RPD, one particular methodology was utilized within the framework, as suggested by Lin & Tu (1995), but can be extended to include other methods (Robinson *et al.*, 2004).

Another contribution from this research is the development of ANN approaches to RPD, as well as system degradation in RPD. Radial basis function neural networks and generalized regression neural networks were explored in RPD. Two different methods of processing response data generated for RPD research were discussed, which can be utilized with combined or crossed array designs. Successful applications of the neural networks and response data processing approaches to highly non-linear problems are

presented. The use of ANNs is appropriate when quadratic regression fails to perform well.

Finally, the contribution of an alternative method of reducing multiple responses to a single dimension was derived. A non subject-matter expert method utilized linear combination techniques that turned multiple response problems into a single dimension for easier RPD analysis.

1.5 Organization of Dissertation

The following is the organization of the dissertation. A summary of current literature pertinent to robust parameter design, artificial neural networks, multiple responses, and factor analysis is provided in Chapter 2. The literature provides a basis upon which these techniques are expanded and applied to RPD.

In Chapter 3, an algorithm is developed to determine doubly robust settings in RPD and an example problem is provided. In this chapter, the use of ANNs and quadratic regression as applied to RPD problems are contrasted. Finally, factor analysis is applied to the response data to reduce the dimensionality of the responses. If factor analysis yields more than one factor, eight linear combination methods are suggested to reduce these factors to a single dimension. A five response problem is explored to demonstrate the use of factor analysis on multiple responses.

In Chapter 4, the technique of doubly robust settings is applied to a current detection algorithm employed by the Air Force. Also, ANNs are implemented in this same detection algorithm to demonstrate their superiority to quadratic regression in highly non-linear problems. Finally, the factor analysis techniques are applied to the four

responses produced from the detection algorithm to show their usefulness over simply summing the normalized response data. Results are summarized and discussed for all techniques.

In Chapter 5, the contributions and conclusions of this research are presented. Also, recommendations for future work are explored.

II. Literature Review

2.1 Overview

Topical areas pertinent to this research are highlighted in this chapter. These areas include robust parameter design, neural networks, multiple response problems, and factor analysis. A brief background on each of the specified areas is given to demonstrate current knowledge in each of their respective fields thus building a foundation on which to further this work into unexplored realms.

2.2 Robust Parameter Design

When performing experiments that contain controllable and uncontrollable parameters, robust parameter design (RPD) is implemented to obtain a desired output value while minimizing the variance caused by the settings of the controls and the various noise in the system (Myers *et al.*, 2002). Genichi Taguchi (1986) first introduced the method of RPD to the United States in the 1980s. Much controversy was raised on Taguchi's approach, but since then, new response surface methods have been developed. These methods are more accepted in the statistical and engineering communities. The use of RPD has extended to a wide array of experimental designs and has been implemented in the practice of many top companies such as AT&T, Ford, and Xerox (Myers & Montgomery, 2002).

Myers & Montgomery (2002) summarize four different focuses of RPD. The first focus involves designing a system that is fairly insensitive to environmental factors once the system becomes operational. The second focus is to design the system to be

insensitive to variability caused when the system becomes operational. The third focus of RPD is to design the system as close as possible to the specifications desired by the user. Finally, the conditions of the system should be set to achieve a target value while minimizing the variance present around the target value.

Primarily, RPD is employed based on the fourth focus presented which achieves a target value while minimizing the variability. To perform RPD, one must understand the variables involved in the system (Brenneman & Myers, 2003). Two types of variables exist: controllable, denoted by x , and uncontrollable (noise), denoted by z . The control variables of the system are those variables the user is able to set. Noise variables are those variables present that the user cannot control and may be known or unknown. RPD is used if noise variable settings produce different outputs of the system when combinations of control settings are selected. Thus the response, Y , is assumed to be a function of the controllable variables and the noise: $Y = f(x, z)$.

The primary interest in the two types of variables lies in the interaction between the two. If the noise variables are independent of the control variables, then the variance of the control variables is constant and the need for RPD is moot. However, if an interaction between the two types of variables exists, then RPD is employed to determine which settings of the control variables should be utilized to minimize the variance. Brenneman & Myers (2003) present Figure 1 below to demonstrate the interaction of control and noise. It is seen that if the control setting is set at “high” then noise has no real effect on the control variables thus the variance is constant for this setting. However, at a setting of “low” for the control variable, significant variance exists in the system and this setting is not desired since the output will be inconsistent.

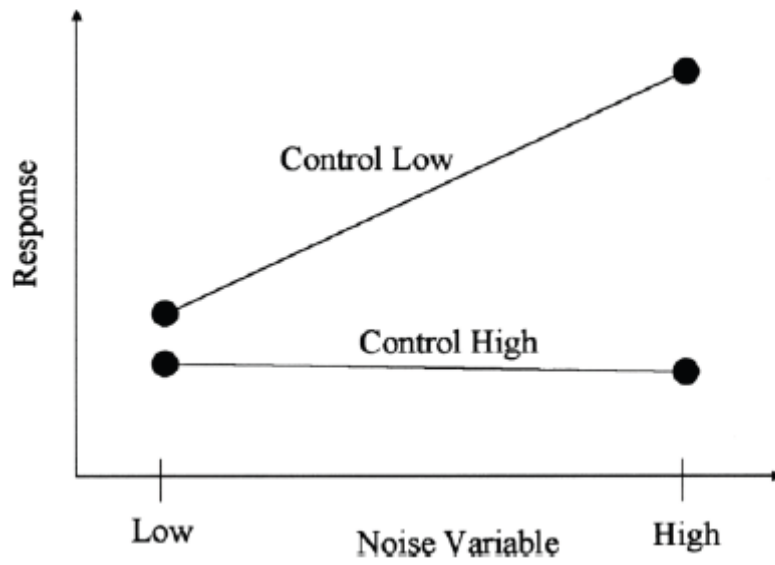


Figure 1. Control and Noise Variable Interaction Plot

Two design approaches exist to perform RPD on a process. The first design, developed by Taguchi, utilizes a crossed array design. The second design, developed by the response surface community, utilizes a combined array design. An in-depth examination into the two designs will be presented in this document as well as comparisons and contrasts.

2.2.1 Crossed Array Design

Taguchi suggested an orthogonal array consisting of control variables to be crossed with an orthogonal array of noise variables, which generated a crossed array design (Myers *et al.*, 2002). The outer and inner array designs can be full or fractional factorial, but an outer array design must be performed at all of the inner array points. For example, an experiment with two control variables and two noise variables, with full

fractional designs on both arrays, consists of $2^2 \times 2^2 = 16$ design points. Figure 2 below displays this example of a crossed array design.

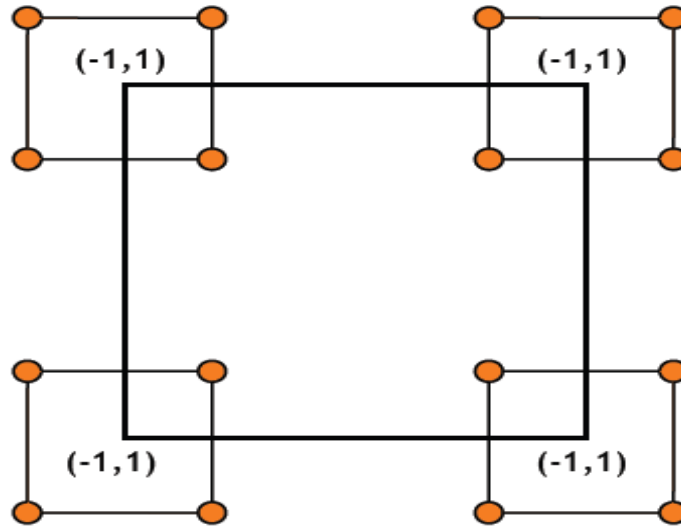


Figure 2. Crossed Array Design (Myers & Montgomery, 2002)

As another example, a three control and two noise variable problem, with full factorials on both the inner and outer arrays, produced a design matrix of size $2^3 \times 2^2 = 32$. The number of control variables increased by one and the number of design points doubled. Thus, this design can lead to a large number of runs if there exists several control and noise variables or if more than two levels are chosen for each factor.

Reduced designs, in terms of resolution, can be used on the inner and outer arrays, but the issue remains that the outer array must be performed at each inner array design point. This particular design allows the user to understand any control by noise interactions that may exist, but limits the ability to understand control by control or noise by noise interactions.

To analyze the data in an experiment, Taguchi suggests the use of a summary statistic on the outer array known as the signal-to-noise ratio (SNR). SNR values summarize the mean and variance into a single statistic. Taguchi presents four different SNRs to use on three different instances. For development of all of the SNR equations, the following quadratic loss function is used (where $y = f(x, z)$):

$$L = E_z (y - t)^2 \quad (2.1)$$

E_z is the expectation operator on the random variable z and t represents the target value on the mean. The different SNR equations are described in the following section and adapted from Myers & Montgomery (2002). The first instance minimizes the response and has a quadratic loss function, $E_z (y - 0)^2$ where t is zero due to minimizing the response (assuming the response is nonnegative). This loss function leads to the following equation:

$$SNR_{(\min)} = -10 \log \sum_{i=1}^n \frac{y_i^2}{n} \quad (2.2)$$

This equation sums the squared errors, divides that number by the number of outer array points, and then sends it through a -10 log transformation. Due to the transformation, the maximum SNR is desired. Using the -10 log (base 10) transformation allows the user to maximize the SNR value despite whether the problem is a minimization, maximization, or target value problem.

The second instance maximizes the response and replaces y_i with $1/y_i$ in Equation (2.2). This allows the quadratic loss function to approach zero as y increases. The resulting SNR is given by the equation:

$$SNR_{(max)} = -10 \log \sum_{i=1}^n \frac{1/y_i^2}{n} \quad (2.3)$$

A target mean value is achieved in the third instance. Two scenarios exist when performing this SNR calculation. The first scenario involves the response mean and variance to be altered independently (Myers & Montgomery, 2002: 541). The control (tuning) factors that have no effect on variance are adjusted to obtain the mean and, thus, the variance is not affected. Once completing this step, the remaining factors are then tuned to maximize SNR, thus, minimizing variance in the system. The resulting SNR equation uses only a transformation of sample variance and is given by:

$$SNR_{(target)} = -10 \log s^2 \quad (2.4)$$

The second scenario exists when the response variance is related to the response mean. The user desires a linear relationship, but this may not always be the case. As before, the control (tuning) factors are set and the remaining factors utilize a maximized SNR value to obtain minimum variance. The resulting SNR value is given by:

$$SNR_{(target)} = 10 \log \frac{\bar{y}^2}{s^2} \quad (2.5)$$

Factor plots can visually assist in the selection of the control variables. Along with plotting the SNR values, the process means can be plotted to determine which control settings are best to use. When performing the analysis, Taguchi's method suggests setting the SNR value at its maximum first, and then choosing the appropriate mean value setting. Using the two plots in conjunction with one another should lead to a favorable answer in achieving a target mean while minimizing variance.

The use of the SNR statistic has been widely criticized largely due to its inability to distinguish between the effects of the mean and variance in the process. Another criticism of the SNR usage is the issue of units. The only unit-less equation is Equation (2.5), SNR_{T_2} . Finally, as discussed previously, crossed array designs can become very large as more controllable and noise factors are added to the design.

2.2.2 Combined Array Design

A combined array design utilizes a smaller number of design points than the crossed array design while continuing to capture the design space of the variables (Montgomery & Myers, 2002). This differs from crossed array designs because every combination of control variables does not need to be tested across every combination of noise variables. Rather, a design is chosen that is intelligent in its construction to test different points in the design region which provide appropriate results. These types of designs are smaller in terms of number of runs compared to crossed array designs.

Response surface methodology utilizing quadratic regression will be used to solve problems with combined array designs. Therefore, when choosing an experimental design for combined array designs, one must consider designs appropriate for second-order models (Myers & Montgomery, 2002). Myers & Montgomery (2002: 304) state that several important properties are necessary in the selection of designs. These four properties are utilized when selecting designs in this research:

1. Result in good fit of the model to the data
2. Give sufficient information for lack of fit
3. Provide an estimate of “pure” experimental error
4. Be cost-effective

Many second-order designs have been established and variations of these designs are continually developed. This research primarily focuses on the central composite design (CCD). First introduced by Box & Wilson (1951), a CCD with k variables uses F factorial points, $2k$ axial points, and n_c center runs (Myers & Montgomery, 2002). The factorial points allow for an estimation of linear and interaction terms, while the axial points estimate quadratic interaction terms and the center points account for estimate of error in quadratic terms (test for curvature). A typical CCD design is depicted in Figure 3 where the axial distance is equal to one ($\alpha = 1$, where α denotes the axial distance).

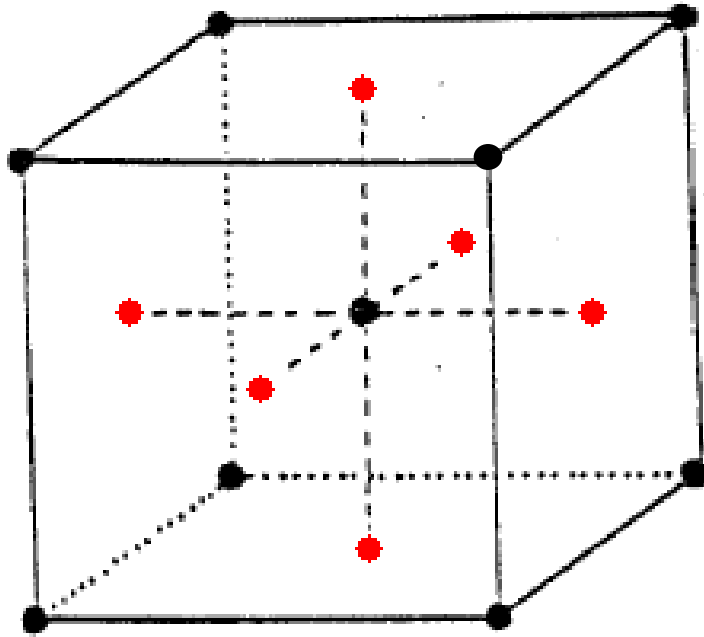


Figure 3. Central Composite Design with Axial Points = 1

Axial points equal to a value of one are shown in Figure 3. This is a special case of CCD known as face-centered. A face-centered cube is used when the design points represent the absolute bounds of the variables settings. Therefore, lower/higher settings than those representing absolute bounds for experimentation cannot be used since

unstable results would occur outside these bounds set by alpha. However, if setting values can be tested outside the low/high settings, then axial points greater than one can be tested. The axial distance varies from 1 to \sqrt{k} .

Other second-order designs exist for use in RPD. If three levels are practical, a 3^k factorial design can be employed. As a special case, the 3^2 factorial is indeed a face-centered CCD. Other designs consist of Box-Behnken, Equiradial, D-Optimal, etc., but only CCD or full factorials are utilized in this research due to their efficiency in developing good quadratic fits and small design size (with the CCD).

2.2.3 Response Surface Methodology

Response surface methodology has extended the ideas of Taguchi to be applicable to RPD. Using response surface methodology, an understanding of the relationship between process mean and variance becomes useful in choosing more appropriate control settings. Combined array designs, discussed in the previous section, are implemented rather than crossed array designs due to their small size and efficiency in providing appropriate quadratic regression fits. However, if the problem is small and inexpensive to test, a crossed array design is more beneficial due to increased sampling of the process.

Two approaches exist to determine the mean and variance models; these are known as the “single model” and the “dual model” approaches. Both methods have been shown to be effective and can be used interchangeably (Robinson *et al.*, 2004). The dual model approach develops a mean model and variance model separately based on collected or historical data (Myers & Carter, 1973). The single model approach differs

from the dual model in that it develops an overall process model from which the mean and variance models are derived.

In the dual model approach, the mean and variance models are created separately from experimental testing. Data is collected on both the mean and the variance of a particular response. By having these two data types, different models are created separately to determine the mean and variance values pertaining to specific control settings. Drawbacks to this approach usually involve the need for a crossed array design, thus, a larger number of runs. However, if the data exists, this approach may prove useful in attaining a more appropriate and exact variance model.

The single model approach applies the response surface methodology technique of developing an overall process model based on collected data and then derives the mean and variance models from the process model. Typically, a second order model is applied to model the interactions between control variables and noise variables, as well as control variables with themselves. Myers & Montgomery (2002) present the equation for the overall process model as:

$$\hat{y}(x, z) = \beta_0 + x' \beta + x' B x + z' \gamma + x' \Delta z + \varepsilon \quad (2.6)$$

The x 's represent the control variables (settings) and the z 's represent the noise variables. The symbol β is a vector of coefficients for the control main effects and γ is the vector of coefficients for the noise main effects. The matrix of coefficients B is the quadratic control effects and Δ is the matrix of coefficients for the control by noise interactions. Finally, ε is distributed normally with a mean of zero and variance, σ^2 , which is estimated by the mean-square error (MSE). This model is “broken down” to

obtain the mean and variance response models. The mean and variance equations are presented below which are adapted from Myers & Montgomery (2002: 563):

$$\text{Mean Response Model: } E_{(z,\varepsilon)}[y(x,z)] = \beta_0 + x'\beta + x'Bx \quad (2.7)$$

$$\text{Variance Response Model: } V_{(z,\varepsilon)}[y(x,z)] = \sigma_z^2(\gamma + \Delta'x)'(\gamma + \Delta'x) + \sigma^2 \quad (2.8)$$

An assumption is made in the variance response model that the variance-covariance matrix of the noise variables given as $\text{cov}(z) = \sigma_z^2 I$. This assumption allowed Myers & Montgomery (2002) to derive the variance response model in Equation (2.8).

The mean response model is directly extracted from the overall process model. When the control variables are set, the same result will be achieved on the average for that particular selection of settings.

The variance model does not offer quite as a direct interpretation as the mean model. However, the model is only in terms of control variables. Only the coefficients of the noise variables and their interactions are used. Although most of the derivation is fairly straightforward matrix algebra, two components are needed to be explained (Myers & Montgomery, 2002). The variance of the noise, σ_z^2 , is related to the coded bounds of the noise variables. Typically this value is assumed to be 1 since the tested bounds of noise variables are between [-1,1]. However, this variance can change based on the bounds of the noise variables thus attaining a value of 2, 1/2, 3, 2/3, etc. This issue requires further research to determine an optimal setting for σ_z^2 . Finally, σ^2 is directly taken as the error for regression given by the overall process model.

The single model approach allows for the use of a combined array design with CCDs. Typically, for RPD, an optimization program is proposed in choosing the control

settings that best achieve the target mean while minimizing variance. Myers & Montgomery (2002) present an optimization problem as suggested by Vining & Myers (1990) which is:

$$\begin{aligned} \min_{x \in D} \quad & V_{(z, \varepsilon)}[y(x, z)] \\ \text{s.t.} \quad & E_{(z, \varepsilon)}[y(x, z)] = m \end{aligned} \tag{2.9}$$

This concludes the development of choosing a design, obtaining a mean and variance model, and establishing the optimization problem for RPD. All that remains is choosing the control settings that optimize Equation (2.9). This problem involves solving the mean and variance models simultaneously and therefore is a dual response problem. Section 2.2.4 addresses different methods proposed in solving the dual response problem that have been applied in RPD research.

2.2.4 Solving the Dual Response Problem

Different versions of Equation (2.9) are used but the general idea is given in the optimization problem presented above (Tang & Xu, 2002; Robinson *et al.*, 2004; Shaibu & Cho, 2009). Myers & Montgomery (2002) present a step-wise approach to solving the optimization problem that is much like the Taguchi approach for solving the dual response problem with SNRs and mean outputs.

First, all possible combinations of settings (at a given step size) are applied to the models, obtaining a response of mean and variance for each combination of settings. The control settings associated with the minimum variance value is chosen, assuming the mean response satisfies the constraint in Equation (2.9). If the constraint is not satisfied, the combination of control settings with the next lowest variance value is chosen. This

procedure is repeated until the mean response constraint in Equation (2.9) is satisfied.

The resultant solution is a unique combination of control variable settings that achieves a target mean with minimum variance across noise in the system.

Myers & Montgomery (2002) also suggest the use of contour plots. However, contour plots are limited to problems consisting of two control variables. The mean contour plot for the two control variables is overlaid with the variance contour plot. This method visually displays the optimal solution if the two are in the same region. However, if the optimal solution is not the same region, once can visually assess the tradeoffs of mean or variance by searching the control variable space.

Myers & Carter (1973) as well as Vining & Myers (1990) suggest optimizing the primary response subject to the secondary response (the constraint) through the use of Lagrangian multipliers. Although the notation is slightly different, the two models (mean and variance) are estimated as:

$$\begin{aligned}\hat{u}_y &= \beta_0 + x' \beta + x' \beta x \\ \hat{\sigma}_y^2 &= \gamma_0 + x' \gamma + x' C x\end{aligned}\tag{2.10}$$

A constraint is added, $x'x = \rho^2$, to restrict the possible search area of the optimal settings to a sphere (where ρ represents the radius of the spherical region). The Lagrangian multipliers are then utilized by associating λ_θ with the mean or variance model and associating λ_ρ with the above constraint (spherical region). Robinson *et al.* (2004) provide an example of solving this optimization problem by minimizing variance and keeping the mean response on a target of 500. Equation (2.11) is minimized over all

possible combinations of λ_θ and λ_ρ to provide \mathbf{x} , which is the optimal set of operating conditions:

$$L = \hat{\sigma}_y^2 - \lambda_\theta(\hat{u}_y - 500) - \lambda_\rho(x'x - \rho^2) \quad (2.11)$$

Del Castillo & Montgomery (1993) took Equation (2.11) and applied the Generalized Reduced Gradient (GRG) algorithm. GRG is utilized because the Lagrangian Multiplier method from Vining & Myers (1990) may not always produce a local optima since only equality constraints are used. GRG allows for the use of inequality constraints more suitable for nonlinear problems. These authors display the effectiveness of this algorithm against the cases of maximizing, minimizing, or even achieving a target value. This method is often preferred due to its built in implementation in common software such as the Microsoft Excel add-in Solver.

Lin & Tu (1995) suggest the incorporation of bias in the primary response in order to avoid forcing the estimated mean response to a particular value. These authors propose minimizing the mean squared error in the optimization problem, Equation (2.9). The three instances that include minimizing, maximizing, and target value are given below respectively (Lin & Tu, 1995; Koksoy, 2008). In all three cases, the MSE value is minimized to provide the solution.

$$\begin{aligned} MSE_{\min} &= \{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)] \\ MSE_{\max} &= -\{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)] \\ MSE_{\text{target}} &= \{\hat{u}_z[\hat{y}(x, z)] - T\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)] \end{aligned} \quad (2.12)$$

Shaibu & Cho (2009) extended the ideas of Lin & Tu (1995) to incorporate target variances into the three models. Lin & Tu assume that a variance of 0 is best in all three

situations as is in most situations of experimental designs and systems. Shaibu & Cho (2009) suggest setting a variance value (S) that the user can live with and this target value (T_s) is added to the variance portion of all three MSE models given above. These MSE equations are:

$$\begin{aligned} MSE_{\min} &= \hat{u}(x) + (\hat{\sigma}(x) - T_s)^2 \\ MSE_{\max} &= -\left[\hat{u}(x) + (\hat{\sigma}(x) - T_s)^2 \right] \\ MSE_{\text{target}} &= (\hat{u}(x) - T)^2 + (\hat{\sigma}(x) - T_s)^2 \end{aligned} \quad (2.13)$$

Copeland & Nelson (1996) point out that the formulation given by Lin & Tu place no restriction on the estimated mean response. Thus, if the estimated mean response values are large and the variances are small, the MSE indications could make the suggested solution far from the minimum variance value. These authors suggest placing a restriction on the MSE search ($\hat{\sigma}^2(x) + \varepsilon$) with the following constraint (given only for the mean target value but applies to minimization/maximization as well):

$$\varepsilon = \begin{cases} (\hat{u}_y - T)^2 & \text{if } (\hat{u}_y - T)^2 > \Delta^2 \\ 0 & \text{if } (\hat{u}_y - T)^2 \leq \Delta^2 \end{cases} \quad (2.14)$$

Tang & Xu (2002) incorporated all these ideas and those of goal programming to derive their own all encompassing approach to the dual response problem. Their objective function and constraints take on the form:

$$\begin{aligned} \min_x \quad & \delta_u^2 + \delta_\sigma^2 \\ \text{s.t.} \quad & \hat{u}_y - w_u \delta_u = T_u^* \\ & \hat{\sigma}_y^2 - w_\sigma \delta_\sigma = T_\sigma^* \\ \text{and} \quad & x'x \leq \rho^2 \quad \text{or} \quad x_l \leq x \leq x_u \end{aligned} \quad (2.15)$$

The weights (w) are user defined and if a rectangular region is used rather than a spherical region, the $\mathbf{x}'\mathbf{x}$ constraint is swapped out. T_u^* and T_σ^* represent the desired target mean and variance values for \hat{u}_y and $\hat{\sigma}_y^2$ respectively. The terms in the objective function, δ_u^2 and δ_σ^2 represent unrestricted scalar variables. These terms multiplied by the weights introduce slackness. One can obtain the target values exactly (set equal to zero), over shoot (> 1) or under shoot (< 1).

This new optimization problem incorporates the techniques of Vining & Myers (1990), Del Castillo & Montgomery (1993), Lin & Tu (1995), and Copeland & Nelson (1996) as special cases depending on the weights of the constraints. For instance, setting w_u and w_σ equal to 1 and T_σ^* equal to 0, yields the following objective function:

$$\delta_u^2 + \delta_\sigma^2 = \left(\hat{u}_y - T_u^* \right)^2 + \hat{\sigma}_y^2 \quad (2.16)$$

This objective function is the same objective function derived from Lin & Tu, Equation (2.12). The other methods can also be obtained through specific weights and target values. This formulation allows one to encompass any established method of solving the dual response problem, or perform goal programming to determine a new set of weights, thus finding a new method of solving this problem.

The different approaches taken to solve the dual response problem for RPD are summarized in Table 1. Work in this document will focus primarily on the Lin & Tu (1995) approach to solving this dual response model. However, the choice of approach is dependent on the application and, as such; the Lin & Tu approach is not necessarily the optimal approach.

Table 1. Current Methodologies to Solve Dual Response Problem in RPD

Author	Method	Objective Function	Constraints	Comments
Myers & Montgomery (MM)	Minimize Variance, Choose Mean or Contour Plots	$\min_x V_{(\varepsilon, z)}[y(x, z)]$	$E_{(\varepsilon, z)}[y(x, z)] = m$	- No tradeoff considered - Contour Plots limited to 2 variables
Vining & Myers (VM)	Lagrangian Multipliers	$L = \hat{\sigma}_y^2 - \lambda_\theta(\hat{u}_y - T) - \lambda_\rho(x'x - \rho^2)$	$x'x = \rho^2$	Difficult calculations
Del Castillo & Montgomery (DM)	Generalized Reduced Gradient	$L = \hat{\sigma}_y^2 - \lambda_\theta(\hat{u}_y - T) - \lambda_\rho(x'x - \rho^2)$	$x'x \leq \rho^2$ Inequality Constraints	Difficult calculations
Lin & Tu (LT)	Mean Squared Error (MSE)	$MSE_{\min} = \{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)]$ $MSE_{\max} = -\{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)]$ $MSE_{\text{target}} = \{\hat{u}_z[\hat{y}(x, z)] - T\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)]$	none	- Uses tradeoffs - No restriction on \hat{u}_z
Shaibu & Cho (SC)	MSE with variance target value	$MSE_{\min} = \hat{u}(x) + (\hat{\sigma}(x) - T_s)^2$ $MSE_{\max} = -\left[\hat{u}(x) + (\hat{\sigma}(x) - T_s)^2\right]$ $MSE_{\text{target}} = (\hat{u}(x) - T)^2 + (\hat{\sigma}(x) - T_s)^2$	$\hat{\sigma}(x) \leq T_s$	- Utilizes target value from variance
Copeland & Nelson (CN)	LT MSE with search restriction	$MSE_{\min} = \{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)]$ $MSE_{\max} = -\{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)]$ $MSE_{\text{target}} = \{\hat{u}_z[\hat{y}(x, z)] - T\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)]$	$\begin{cases} (\hat{u}_z - T)^2 & \text{if } (\hat{u}_z - T)^2 > \Delta^2 \\ 0 & \text{if } (\hat{u}_z - T)^2 \leq \Delta^2 \end{cases}$	- Reduces distance \hat{u}_z can move from target value
Tang & Xu (TX)	Quadratic Optimization Problem	$\min_x \delta_u^2 + \delta_\sigma^2$	$\hat{u}_y - w_u \delta_u = T_u^*$ $\hat{\sigma}_y^2 - w_\sigma \delta_\sigma = T_\sigma^*$ $x'x \leq \rho^2$ or $x_l \leq x \leq x_u$	- Encompasses all above methods through “special cases” - Weighting can be subjective

The settings found to be optimal based on the dual response model should not only achieve a desired mean output of the system, but should also be robust to any noise properly modeled in the system. The following section briefly discusses “robustness.”

2.2.5 Robustness

A robust design is a design that implements a particular set of settings that provide good mean performance and is insensitive to uncontrollable sources or variables that cause variation (Sanchez, 1994). The key word in the previous definition is “good.” Determining good solutions is difficult due to personal bias and understanding of the process or system. “Robust” will be used in this document to mean control variable settings that are insensitive to noise in the system. This means that the variance of the process is relatively low across the noise space under the “robust” settings and that the mean is close to its target value.

Each method in Table 1 presents a solution that is the most robust, according to the method’s formulation. However, although the settings are robust, they may provide weaker mean (and variance) responses than desired (unless explicitly expressed as a target mean or variance). Most of the methods “search” the solution space to provide a tradeoff between increases in expected mean (assuming maximizing) while maintaining little change in variance. However, the further one moves away from the minimum variance value, the less robust the solution set becomes. The goal of RPD is not to necessarily provide the best mean for the given situation, but rather to provide a consistent mean for future implementation of the process or system with the existence of uncontrollable noise variables (Myers & Montgomery, 2002).

2.3 Artificial Neural Networks

Methodology has been discussed on how to approach an RPD problem, determine optimal solutions, and determine robustness using quadratic regression. The success of this analysis is based on how well one is able to fit training data with a regression model (achieving significance, denying lack of fit, obtaining high r-squared values, etc.).

Neural networks can be used to fit the regression models, rather than traditional linear/quadratic techniques, allowing for a more nonlinear and hopefully a better fit.

Artificial neural networks (ANNs) were established with the notion that the human brain could be mimicked by an engineering design (Kuncheva, 2004). These ANNs resemble the biological cognitive systems with their ability to “learn” data and patterns through the use of supervised training for parameter adjustment in the model. Many types of ANNs are employed in practice today, each with different learning rules and differences in the calculation of outputs for each specific neural network.

2.3.1 ANN Classification and Regression

ANNs are used for either classification purposes or regression analysis. Loeffelholz *et al.* (2009) successfully demonstrated the use of four different ANNs to classify a winner in an NBA basketball game based simply on box score data. The results obtained from these authors showed remarkable improvement in accuracy over the “experts” in the field of basketball while using the simplest form of data collected in the sport. In classification, ANNs can be superior to techniques such as discriminant analysis, factor analysis, or principal component analysis due to their ability for non-linear fits.

In this research, rather than determining a two or three class output system, ANNs are used to fit regression type models. Myers & Montgomery (2002) claim that quadratic regression will fit most real world application problems but when this is not true, an alternative formulation is necessary to model those problems not captured well using quadratic regression. For instance, Davis (2009) applied RPD to a hyper-spectral imagery problem consisting of four outputs. For two of the four outputs, very low R^2 values and significant lack of fit was present through the use of quadratic regression. This result warrants the use of an alternate method, such as ANNs, to properly fit data.

As noted previously, different neural networks have been developed to model difficult problems. The neural networks chosen for this research are Radial Basis Function Neural Networks (RBFNNs) and Generalized Regression Neural Networks (GRNNs). These networks were chosen based on their applicability to regression analysis as well as a positive personal experience of the author through applying these ANNs to real world problems.

2.3.2 Radial Basis Function Neural Networks

To begin the discussion on RBFNNs, the architecture will be presented followed by the underlying mathematics behind RBFNNs. The RBFNN is constructed using a layer of input nodes, a single hidden layer, and an output layer. The input layer is related to the number of features, or in the case of regression, the number of (functional) independent variables. The number of nodes in the hidden layer is equal to the number of training exemplars in the input layer. For example, if a CCD consisting of 23 runs where 15 runs were allocated to the training set and the remaining eight were withheld for

testing, the hidden layer of the RBFNN would consist of 15 hidden nodes. Finally, the output layer is related to the number of outputs in the system. In this research of using RBFNNs for regression, this output layer will consist of a single node if examining a single response problem. Depending on the formulation of the RBFNN output, two bias layers can also exist. An initial bias layer can be applied on the input and another bias layer on the output.

A RBFNN (biases not shown) is depicted in Figure 4. \underline{X}_i represents the input features or the variables for regression. Every exemplar in the input layer is passed to each node in the hidden layer. Each hidden layer node contains a basis function (h_i) which is weighted (w_i) to the output node. The output layer sums (Σ) all the weighted hidden layer values to obtain the output value, y , for each combination of input variables.

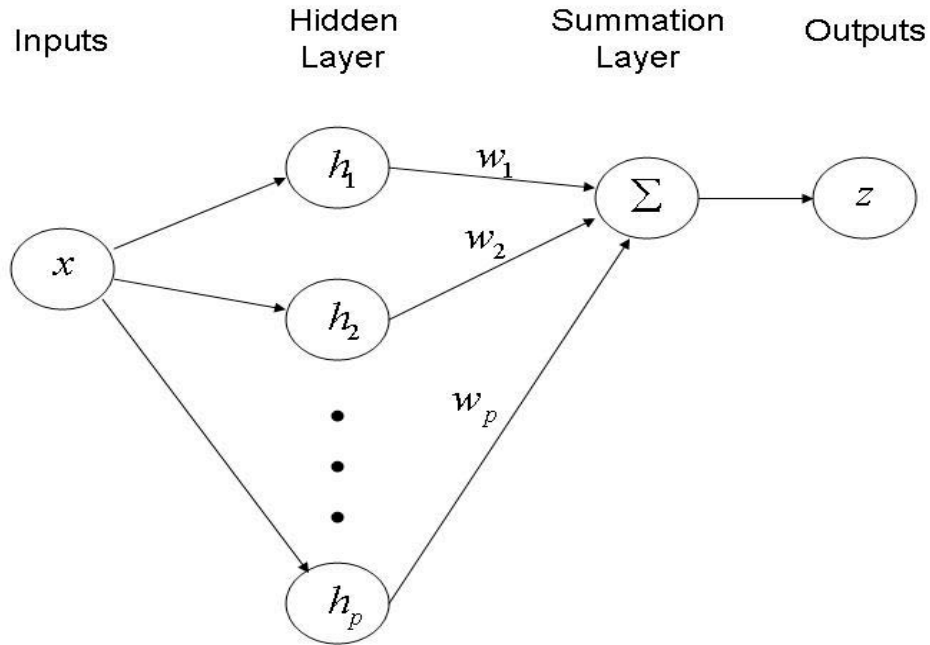


Figure 4. RBFNN with Single Output

Typically the Gaussian function is applied to the hidden layer nodes and is given by:

$$h_i(\underline{x}) = \exp\left(\frac{-(\underline{x} - \mu_i)^T(\underline{x} - \mu_i)}{2\sigma_i^2}\right) \quad (2.17)$$

For notational purposes, \underline{x} represents the exemplar sent through the network, μ_i is the i th center, and σ_i represents the i th spread. To explain this network in simple terms, the classification application is first discussed. The exemplar is sent through the network and its distance from the centers (or trained exemplars) is calculated and those firing closest to a particular center score a value closest to that used in training. For instance, if two centers with input values of [1 2] and [4 5] representing class 1 and class 2, respectively, are trained in an RBFNN, a new exemplar with input values of [1 2] is most likely to fire closest to center [1 2] and be labeled a class 1 node.

In terms of regression, the same principle is applied. Each new input value is measured against all trained values (hidden nodes) to determine their “distance” from each node. The new input value is then assigned an output value closely resembling the output value for the hidden node with the closest activation. Bias and weights are incorporated to allow the output values to vary around the actual output value of the hidden node, thus leading to a function rather than discrete points as used in classification purposes.

Training in an RBFNN is simpler and quicker than other networks (such as Feed-Forward). To train the network, each exemplar is fed through the hidden nodes, one at a

time, obtaining an output. A hidden weight is obtained and the overall weight of the network is adjusted. This is done by the following equation:

$$w_i(n+1) = w_i(n) + \eta(t - y)z_i \quad (2.18)$$

w_i represents the weights of the networks, n is the iteration number, η is the step size, t acts as the target value, y represents the network output, and finally $z_i \equiv h_i(\underline{x})$.

Once again, each exemplar is sent through the network and the weights are updated. This process is continued until an appropriate total error is reached, thus indicating a well trained RBFNN. Since only a linear output layer is used outside the hidden layer, Wasserman (1993) notes that the RBF is guaranteed to converge to a global minimum (as compared to other networks that can be trapped in local minimums) but the network can be extremely large dependent on the number of training exemplars used.

Once a RBFNN is trained, new exemplars can be processed through the network to obtain an appropriate output value. In the literature, Duda *et al.* (2001), Looney (1997), and Wasserman (1993) present the following equation to be used in calculating an output value for a given input:

$$z^{(i)} = \sum_{j=1}^p w_j \exp \left(-\frac{1}{2\sigma_j^2} \sum_{k=1}^n (x_k^{(i)} - \mu_k^{(j)})^2 \right) \quad (2.19)$$

For exemplar i , this formula assumes that p centers exist for n features (or input variables). $x_k^{(i)}$ represents the k th input feature/variable of the new exemplar and $\mu_k^{(j)}$ represents the k th component of the j th center. The distance of the new exemplar from the center of all trained exemplars is calculated. Here σ represents the spread and

w_j is equal to the weight of the j th node. In regression terms, this will provide the expected output value for any given input variable terms.

In this research, MATLAB® is employed by making calculations easier for large scale problems. MATLAB® uses a slightly different formulation for determining the output value of a RBFNN. The formulation used is represented as:

$$z^{(i)} = \sum_{j=1}^p w_j \exp \left(- \sum_{k=1}^n (b_j^1 (x_k^{(i)} - \mu_k^{(j)}))^2 \right) + b^2 \quad (2.20)$$

In this equation, the initial bias term (b_j^1) represents MATLAB's® interpretation of applying the spread in the equation. The bias term is calculated as .8326/spread rather than using one half the squared values as in Equation (2.19). Also, a second bias term is added on the end of the equation to represent a linear layer bias term. This simply shifts the output value up or down by the specified amount.

For a more in-depth examination of RBFNNs, Duda *et al.* (2001), Looney (1997), and Wasserman (1993) are appropriate texts. These authors also provide deeper insight into the origination of RBFNNs, a deeper understanding of the training approach, as well as other details of RBFNNs.

2.3.3 Generalized Regression Neural Networks

Generalized Regression Neural Networks (GRNNs) belong in the same class as the RBFNNs and are useful in terms of non-linear regression (Wasserman, 1993). The architecture is similar but calculations differ in terms of training. More specifically, there exists no training in the GRNN. Response values are directly calculated from the network with the weights directly related to the response values. GRNNs are extremely

useful compared to other neural networks due to their ability to converge on a function with little training data.

Specht (1991) applied the ideas of the Normal Distribution to create the formulation of the GRNN. The formula, where $D_i^2 = (X - X^i)^T (X - X^i)$, is given as:

$$\hat{Y}(X) = \frac{\sum_{i=1}^n Y^i \exp\left(-\frac{D_i^2}{2\sigma^2}\right)}{\sum_{i=1}^n \exp\left(-\frac{D_i^2}{2\sigma^2}\right)} \quad (2.21)$$

The scalar function, D_i^2 , calculates the distance from the new exemplar (X) from the centers of all the “training” exemplars (X^i). The spread value, σ^2 , remains defined as the same for RBFNNs. Finally, Y^i represents the weights in the network which is extracted from the outputs of the “training” exemplars. This feature distinguishes the GRNN from the RBFNN in that the weights do not need to be calculated or updated in the GRNN.

To remain consistent with notation in the RBFNN, the GRNN formula can be re-written as:

$$z = \frac{\sum_{j=1}^p w_j \exp\left(-\sum_{k=1}^n (b_j^1(x_k - u_k^{(j)}))^2\right)}{\sum_{j=1}^p \exp\left(-\sum_{k=1}^n (b_j^1(x_k - u_k^{(j)}))^2\right)} \quad (2.22)$$

The features/variables of the new test point are represented by x with its output as z . Weights for each “training” exemplar are represented as w and are taken from the response for each feature setting. The centers of the “training” exemplars are $\mu_k^{(j)}$ and all remaining terms were defined in RBFNNs. The GRNN based on Equation (2.22) is

depicted in Figure 5. Originally derived from Specht (1991), this figure was modified.

α represents the numerator while β represents the denominator of Equation (2.22).

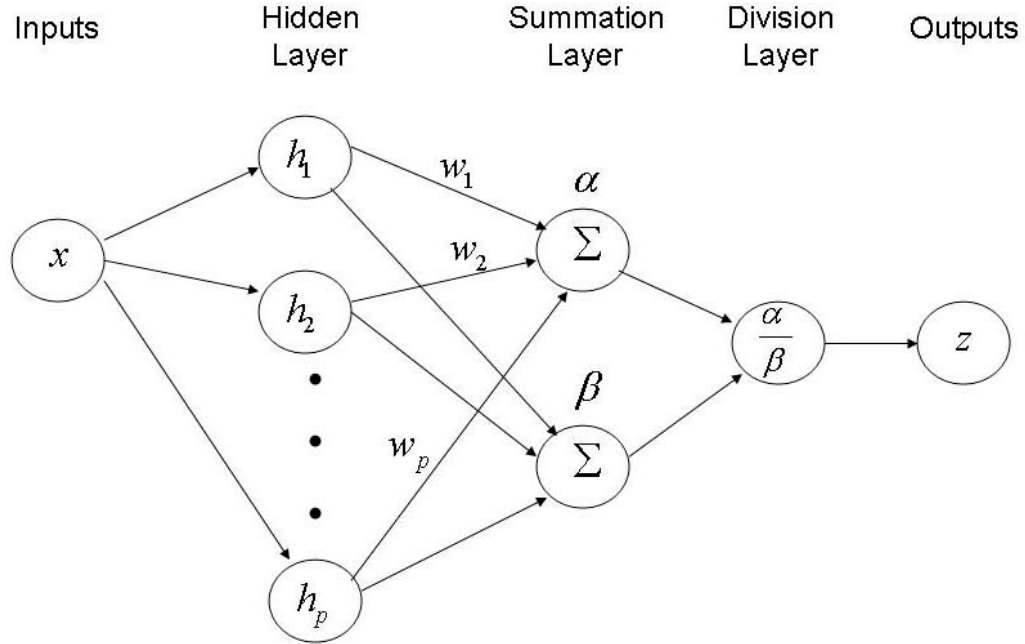


Figure 5. Generalized Regression Neural Network

This regression allows for direct application into problems involving numerical data (Specht, 1991). GRNNs calculate quickly since weights do not have to be calculated and updated separately. For more information on the construction of GRNNs, the reader is referred to Wasserman (1997) or Specht (1991).

2.4 Multiple Responses

Many real-world application problems involve the use of multiple responses rather than a single response. Often the control variables will adjust more than one response differently with the same chosen settings for those variables. Traditional research (Myers & Montgomery, 2002; Robinson *et al.*, 2004) focuses on single response

problems in RPD. Several approaches have been taken to solve multiple response RPD problems and are presented in the following sections.

2.4.1 Weighting the Responses

Perhaps the simplest method of solving the multiple response problems is to assign weights to each of the response values to form a linear combination (Koksoy, 2008), i.e., $\hat{y} = \sum_{i=1}^m w_i \hat{y}_i$ where w_i is the weight of the i^{th} response of m responses. Now optimization is performed relative to \hat{y} . One difficulty with this technique is selecting appropriate values for the weights. The author indicates that expert opinion is usually elicited for the weights. Such methods could lead to ambiguity.

Combating this problem, Decision Analysis (Kirkwood, 1997) can be applied through the use of value models. Once again, an expert must be utilized, but mathematical procedures are used to obtain appropriate weights for each of the response values. Also, this approach allows the user to change weights and see the changes instantly.

Kuhnt & Erdbrugge (2004) extend the idea of using weights by applying a loss function. These authors apply the loss function to multi-response problems to minimize the overall expected loss when applying different combinations of weights to all the responses. Graphs are utilized to show the expected loss values at different settings of the weights.

2.4.2 Visualizing the Responses

For a small number of responses, contour plots can be generated for the response functions (Lind *et al.*, 1960). Overlaying the contour plots of each of the responses highlights areas in which the settings of the control variables prove optimal. This approach becomes increasingly difficult when more responses are added to the problem. Also, another downside may be the lack of good areas found if the responses differ quite dramatically across the setting space of the control variables.

2.4.3 Desirability Functions

Desirability functions can be used to determine the optimal settings for the control variables. This approach is similar to weighting the responses. Derringer & Suich (1980) first proposed the idea of desirability functions by converting each response into its own desirability function that covers the range of zero to one. A value of one represents the response achieving its goal while a value of zero indicates the response is outside the specified acceptable region of interest. The scores for each response are multiplied together and taken to the m th root, where m represents the number of responses. This is represented as:

$$D = \sqrt[m]{d_1 d_2 \cdots d_m} \quad (2.23)$$

The desirability function used can vary (Zandieh *et al.*, 2009; Chang, 2006; Derringer & Suich, 1980; Harrington, 1965) depending on the user's preference. One such function is the exponential function. Myers & Montgomery (2002) adapt the functions presented by Derringer & Suich (1980) creating three desirability functions based on maximizing, minimizing, or achieving a middle ground value. T represents the

target value, L and U are the lower and upper limits of the responses, and r represents how important achieving the target value is. A value of $r = 1$ causes the function to be linear, $r > 1$ places a larger emphasis on achieving the target values while $0 < r < 1$ puts less emphasis on achieving the target value. The equations presented represent the desirability functions for when one maximizes, minimizes, or finds some target value respectively:

$$d_{(\max)} = \begin{cases} 0 & y < L \\ \left(\frac{y-L}{T-L} \right)^r, & L \leq y \leq T \\ 1 & y > T \end{cases} \quad (2.24)$$

$$d_{(\min)} = \begin{cases} 1 & y < T \\ \left(\frac{U-y}{U-T} \right)^r, & T \leq y \leq U \\ 0 & y > U \end{cases} \quad (2.25)$$

$$d_{(\text{target})} = \begin{cases} 0 & y < L \\ \left(\frac{y-L}{T-L} \right)^{r_1}, & L \leq y \leq T \\ \left(\frac{U-y}{U-T} \right)^{r_2}, & T \leq y \leq U \\ 1 & y > T \end{cases} \quad (2.26)$$

Zandieh *et al.* (2009) utilized Equations (2.24)-(2.26) to construct their optimization problem including a constraint. Zandieh *et al.* presented their optimization problem, for k responses, and obtained the following mathematical model:

$$\begin{aligned} \max_x \quad & D = \sqrt[k]{d_1(y_1) \times d_2(y_2) \times \dots \times d_k(y_k)} \\ \text{s.t.} \quad & L(x_h) \leq x_h \leq U(x_h) \end{aligned} \quad (2.27)$$

These authors suggest optimizing Equation (2.27) through the use of search algorithms such as Genetic Algorithms, Tabu Search, and Simulated Annealing. The authors found Simulated Annealing to perform the best in terms of multiple response RPD problems.

Chang (2006, 2008) follows a similar approach to the desirability functions above by implementing the use of a back-propagation network (BPN). Chang's procedure consisted of modeling the different response models through a BPN, evaluating the chosen design space of levels, and using exponential desirability functions and Simulated Annealing to determine settings for the parameters.

2.4.4 MSE and PCA

Koksoy (2008) utilizes the Lin & Tu (1995) methodology (previously described in Section 2.2.4) which involves the use of mean squared error to combine the mean and variance models. However, Koksoy assumes that multiple responses exist; he proposes the following optimization problem for $i = 1, 2, \dots, r$ responses:

$$\begin{aligned} \min_{x \in R} \quad & MSE(x)_{j, (i \neq j)} \\ \text{s.t.} \quad & MSE(x)_i = MSE(x)_{i0} \end{aligned} \tag{2.28}$$

Previously, the region (R) was defined as $x'x = \rho^2$, which applies here as well if a spherical region is used. This method optimizes the appropriate objective function, $MSE(x)_{j, (i \neq j)}$, while setting the values for the remaining MSE functions ($MSE(x)_{i0}$). Koksoy presents a two response problem and solves this by incrementing $MSE(x)_{20}$ at fixed iterations and optimizing $MSE(x)_1$ at each iteration. This approach leads to a table of alternative solutions, or a portfolio of solutions, allowing the user to examine tradeoffs

of choosing one pair of settings over another in terms of changes in the different MSE values. Koksoy (2008) solves this new optimization problem through the use of the Generalized Reduced Gradient (GRG) method developed in nonlinear programming.

Su & Tong (1997) apply Principal Component Analysis (PCA) to examine correlation among responses and utilize the component scores rather than raw responses. These authors apply PCA to the crossed array design approach as suggested by Taguchi (1986). The raw responses are transformed into principal component scores and those scores kept are based on eigenvalues scoring higher than one. Factor plots are used on the component scores (much like the factor plots in SNR) to determine optimal settings for each control factor in the experiment. This method proved to reduce the dimensionality of the problem and decrease the impact of its complexity.

Ideas from these two methods will be applied in Chapter 3 and 4 to formulate a new way of examining multiple response problems. First, in a similar fashion to Koksoy (2008), the Lin & Tu (1995) methodology will be applied to various single and multiple response problems in RPD. Also, much like the PCA approach of Su & Tong (1997), Factor analysis will be implemented and used in a combined array design rather than the Taguchi method of SNRs and crossed array designs. The next section details factor analysis and how it is applied in this research.

2.5 Factor Analysis

Factor analysis is a data reduction technique that attempts to discover underlying factors that link two or more variables with one another (Dillon & Goldstein, 1984). This analysis takes seemingly unrelated variables and finds some linear combination to

combine them, thus determining commonalities between factors. In classification problems, factor analysis helps group similar classes with one another allowing for easier differentiation between classes. Factor analysis allows one to work in a much smaller dimensioned set by working with factor scores as opposed to raw data sets.

2.5.1 Factor Analysis and PCA

Factor analysis and principal component analysis (PCA) are similar with one key difference. That key difference is the explanation of the variance. PCA assumes that the total variance of the variables is included in the components which allows for no error variance. With factor analysis, an error variance is assumed since the commonalities are estimated. Factor analysis looks for common or shared variation rather than attempting to account for all the total variation (Dillon & Goldstein, 1984).

2.5.2 Mathematical Model of Factor Analysis

To mathematically represent factor analysis, the following algebraic representation is used:

$$\begin{aligned}
 X_1 &= v_{1(1)}CF_{(1)} + v_{1(2)}CF_{(2)} + \dots + v_{1(m)}CF_{(m)} + e_1 \\
 X_2 &= v_{2(1)}CF_{(1)} + v_{2(2)}CF_{(2)} + \dots + v_{2(m)}CF_{(m)} + e_2 \\
 &\vdots \\
 X_p &= v_{p(1)}CF_{(1)} + v_{p(2)}CF_{(2)} + \dots + v_{p(m)}CF_{(m)} + e_p
 \end{aligned} \tag{2.29}$$

This representation assumes that one has m common factors on p variables. The number of common factors must be less than the number of variables ($m < p$).

2.5.3 Extracting Factors

Several methods exist for extracting factors from the data set. Dillon & Goldstein (1984) summarize several of the given methods which are presented in the Table 2. The primary difference between the methods is that different factor solutions are obtained based on the method used. Therefore, the authors describe that certain methods should be utilized based on the sample size, number of variables, and variation among variables.

Table 2. Types of Factor Extraction Methods

Principal Components
Principal Factor
Minimum Residual
Image
Alpha
Maximum Likelihood
Canonical Maximum Likelihood

The research presented in this document will use the principal components method for obtaining factors and calculating factor scores. This method maximizes the variance accounted for, which is set by the user. The eigenvalues are used to explain the amount of total variation by each factor.

2.5.4 Factor Rotation and Factor Scores

Along with factor analysis, a varimax rotation is often applied to the data set to obtain a simpler structure for the factors. Thurstone (1947) presented the idea of factor

rotation to allow for easier interpretation, particularly graphically. The varimax rotation takes the variance of the squared factor loadings and attempts to maximize their sum. The primary use of factor rotation is simple interpretability (Dillon & Goldstein, 1984; Anderson, 2003).

Factor (rotated) scores are estimated and represent the location of the observation in terms of the space projected by the factors (Dillon & Goldstein, 1984). These scores allow for a graphical representation of how the different observations lie on a coordinate axis, allowing one to find clusters or links between observations visually. Once the scores are estimated, they can be used as the new response data for the design matrix.

2.5.5 Summary

Factor analysis is utilized in this research to reduce the dimensionality of multiple response problems. Rather than examining the independent variables, the responses are examined in attempt to find common factors to allow for grouping of different responses. This grouping assists in reducing the dimensionality of the multiple responses. Also, factor scores are obtained for each of the new factors which are interpreted as new responses to be analyzed. Performing this analysis allows the user to understand underlying correlations or similarities between different responses that may not have otherwise been seen in the problem.

III. Methodology

3.1 Overview

Methodology employed in this research is covered in this chapter. First, RPD is demonstrated using the Lin & Tu (1995) methodology for choosing optimal settings in problems containing noise variables. Next, a new methodology is developed to determine alternative settings designed to guard against possible system degradation in the future. These are known as “doubly robust” settings.

Artificial neural networks (ANNs) are introduced as an alternative to quadratic regression in RPD for more complex (non-linear) situations. Two examples are provided to compare/contrast settings obtained through quadratic regression and ANNs. Following this, a framework is developed to determine doubly robust settings using ANNs.

Finally, factor analysis is implemented for multi-response problems to reduce dimensionality of the outputs to a single dimension without the use of subject matter experts.

3.2 Robust Parameter Design

As discussed in Chapter 2, RPD is performed using a crossed array design combined with signal to noise ratio values (Taguchi, 1990) or by implementing a combined array design and using response surface methodology (Myers & Montgomery, 2002). Both crossed array and combined array designs were utilized in this research but focused on response surface methodology techniques due to their ability to obtain control

by control variable interactions. These interactions allowed for better understanding of the true nature of the design space's effects on the solution space.

To conduct RPD, the single model approach was used to create an overall response model, also known as a process model. For a problem with p control variables (x) and q noise variables (z), this equation is:

$$\hat{y}(x, z) = \beta_0 + x' \beta + x' Bx + z' \gamma + x' \Delta z + \varepsilon \quad (3.1)$$

β are the coefficients of the control variables, γ are the coefficients of the noise variables, B is the matrix of coefficients for control by control interactions and Δ is the matrix of coefficients for control by noise interactions. Finally, ε represents the error and is distributed normally with a mean of zero and variance, σ^2 , which is estimated by the mean squared error. This process model is obtained through simple regression. For the coefficients (assuming $X'X$ is invertible), the formula $(X'X)^{-1}(X'Y)$ is implemented where X represents the design matrix (from the crossed or combined array) of both the control and noise variables and Y represents the responses obtained during experimentation. From this process model, the mean and variance models can be computed directly. These two models are given by:

$$E_{(z, \varepsilon)}[\hat{y}(x, z)] = \beta_0 + x' \beta + x' Bx \quad (3.2)$$

$$V_{(z, \varepsilon)}[\hat{y}(x, z)] = \sigma_z^2 (\gamma + \Delta' x)' (\gamma + \Delta' x) + \sigma^2 \quad (3.3)$$

After computing these two models, an optimization problem is solved to minimize the variance subject to achieving a target mean. Solving this dual response problem requires choosing an approach outlined in Table 1. The Lin & Tu (1995) method was selected for this research, which henceforth will be known as the LT solution. As a

reminder, the LT solution is obtained in using one of three different objective functions in a *minimization* framework, depending on the problem type.

A minimization problem (*min*) requires the response to be nonnegative since the mean segment of the MSE equation is squared. An assumption is made that a response which equals zero is desired. For maximization problems (*max*), the response must also be nonnegative with no restriction on its upper bound. If the response can be negative or a specific value other than zero (for *min*) or infinity (for *max*) is desired, *target* value problems need to be implemented.

$$\begin{aligned}
MSE_{\min} &= \{E_{(z,\varepsilon)}[\hat{y}(x,z)]\}^2 + V_{(z,\varepsilon)}[\hat{y}(x,z)] \\
MSE_{\max} &= -\{E_{(z,\varepsilon)}[\hat{y}(x,z)]\}^2 + V_{(z,\varepsilon)}[\hat{y}(x,z)] \\
MSE_{\text{target}} &= \{E_{(z,\varepsilon)}[\hat{y}(x,z)] - T\}^2 + V_{(z,\varepsilon)}[\hat{y}(x,z)]
\end{aligned} \tag{3.4}$$

This concludes the discussion of a standard methodology for finding the optimal robust settings. The robust settings obtained here represent settings that are hopefully insensitive to the noise variables in the system and obtain the target mean. The next section presents a small example problem to demonstrate the RPD methodology.

3.2.1 RPD Example Problem

To demonstrate the RPD methodology, a small textbook problem was adapted from Myers & Montgomery (2002: 566). The problem involved a Semiconductor process which contained two controllable variables and three noise variables. It was desired that the system response be minimized. All settings are presented as coded terms since their natural values are of no interest in this demonstration. The text used a

fractional CCD consisting of 23 runs, including center points to allow for estimation of curvature. Table 3 displays the problem data.

Table 3. Semiconductor Example Problem

Run Number	x1	x2	z1	z2	z3	y
1	-1	-1	-1	-1	1	44.2
2	1	-1	-1	-1	-1	30
3	-1	1	-1	-1	-1	30
4	1	1	-1	-1	1	35.4
5	-1	-1	1	-1	-1	49.8
6	1	-1	1	-1	1	36.3
7	-1	1	1	-1	1	41.3
8	1	1	1	-1	-1	31.4
9	-1	-1	-1	1	-1	43.5
10	1	-1	-1	1	1	36.1
11	-1	1	-1	1	1	22.7
12	1	1	-1	1	-1	16
13	-1	-1	1	1	1	43.2
14	1	-1	1	1	-1	30.3
15	-1	1	1	1	-1	30.1
16	1	1	1	1	1	39.2
17	-2	0	0	0	0	46.1
18	2	0	0	0	0	36.1
19	0	-2	0	0	0	47.4
20	0	2	0	0	0	31.5
21	0	0	0	0	0	30.8
22	0	0	0	0	0	30.7
23	0	0	0	0	0	31

Quadratic regression was performed on the data in Table 3 to obtain the process model, Equation (3.5), with each coefficient estimate rounded to the second decimal.

The error term for regression was calculated as $\varepsilon \sim Normal(0, \sqrt{.9526})$.

$$\begin{aligned} \hat{y}(x, z) = & 30.37 - 2.92x_1 - 4.13x_2 + 2.6x_1^2 + 2.18x_2^2 + 2.87x_1x_2 + 2.73z_1 - 2.33z_2 \\ & + 2.33z_3 - 0.27x_1z_1 + 0.89x_1z_2 + 2.58x_1z_3 + 2.01x_2z_1 - 1.43x_2z_2 + 1.56x_2z_3 + \varepsilon \end{aligned} \quad (3.5)$$

Computing Equations (3.2) and (3.3) based on Equation (3.5) yielded the following mean and variance models:

$$E_{(z, \varepsilon)}[\hat{y}(x, z)] = 30.37 - 2.92x_1 - 4.13x_2 + 2.6x_1^2 + 2.18x_2^2 + 2.87x_1x_2 \quad (3.6)$$

$$V_{(z, \varepsilon)}[\hat{y}(x, z)] = 19.26 + 6.4x_1 + 24.9x_2 + 7.52x_1^2 + 8.52x_2^2 + 4.42x_1x_2 \quad (3.7)$$

The overall process model required the response to be minimized thus the LT equation used is:

$$MSE_{\min} = \{E_{(z,\varepsilon)}[\hat{y}(x, z)]\}^2 + V_{(z,\varepsilon)}[\hat{y}(x, z)] \quad (3.8)$$

Due to the small number of control variables, all possible setting combinations between -1 and 1 with a step size of .01 were tested. Recall, these values were in coded terms. All possible combinations were applied to Equation (3.8) to produce an LT solution for each combination. The minimum LT value was chosen and the associated control variable settings represented the optimal robust pair of settings for this problem, as shown in Table 4.

Table 4. LT Settings and Solution to Semiconductor Problem

x1	x2	Est. Mean	Est. Std Dev	Est. LT
0.13	0.71	28.463	6.53	852.74

The results obtained using the LT formulation differ from results obtained by Myers & Montgomery (2002). As shown in Table 1, these authors suggest minimizing the variance and then choosing the associated mean value or using contour plots to locate an appropriate optimal solution. Figure 6 displays the contour plot for the mean model and Figure 7 depicts the contour plot of the variance model (in terms of standard deviation). Figure 8 overlays these two plots showing the region of settings (highlighted oval) for the control variables, as suggested by Myers & Montgomery. These authors solved the optimization problem of minimizing the variance subject to the mean response being less than or equal to 30. Their optimal point was in the region centered around [0.25 , 0]. The LT solution is compared to the Myers & Montgomery solution in Table 5.

Table 5. M&M Solution and LT Solution to Semiconductor Problem

Approach	x1	x2	Est. Mean	Est. Std Dev	Est. LT
M & M	0.25	0	29.797	4.62	909.22
LT	0.13	0.71	28.463	6.53	852.74

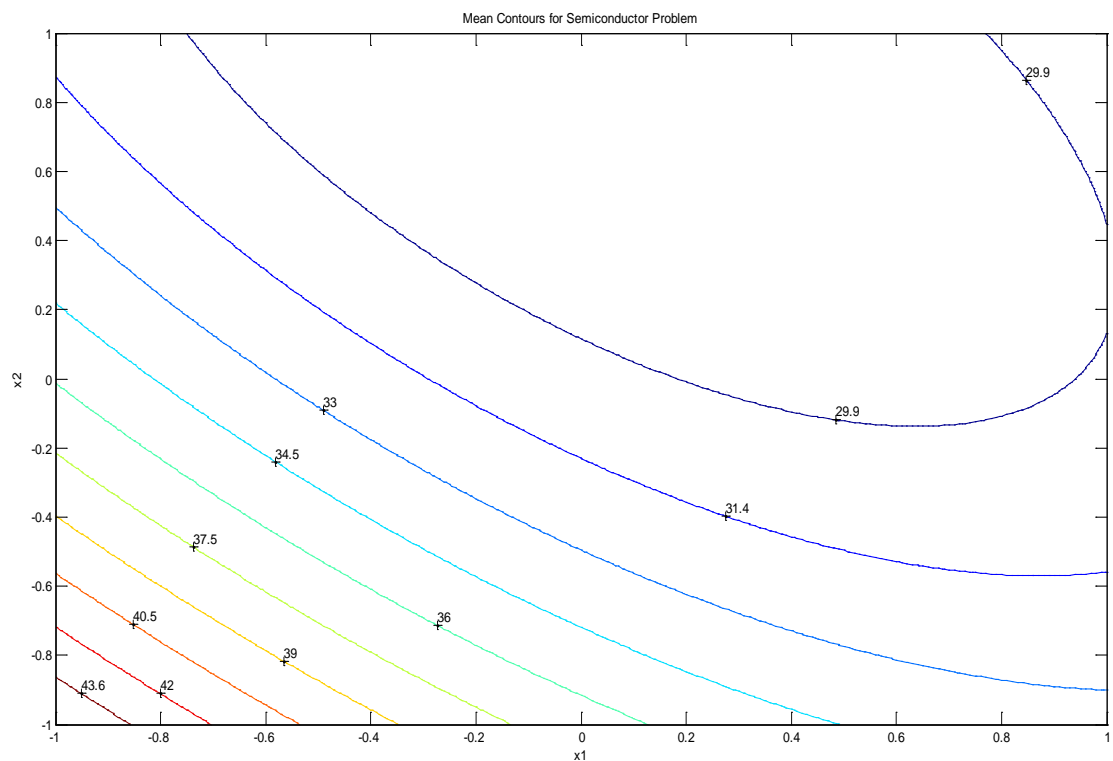


Figure 6. Mean Contour Plot

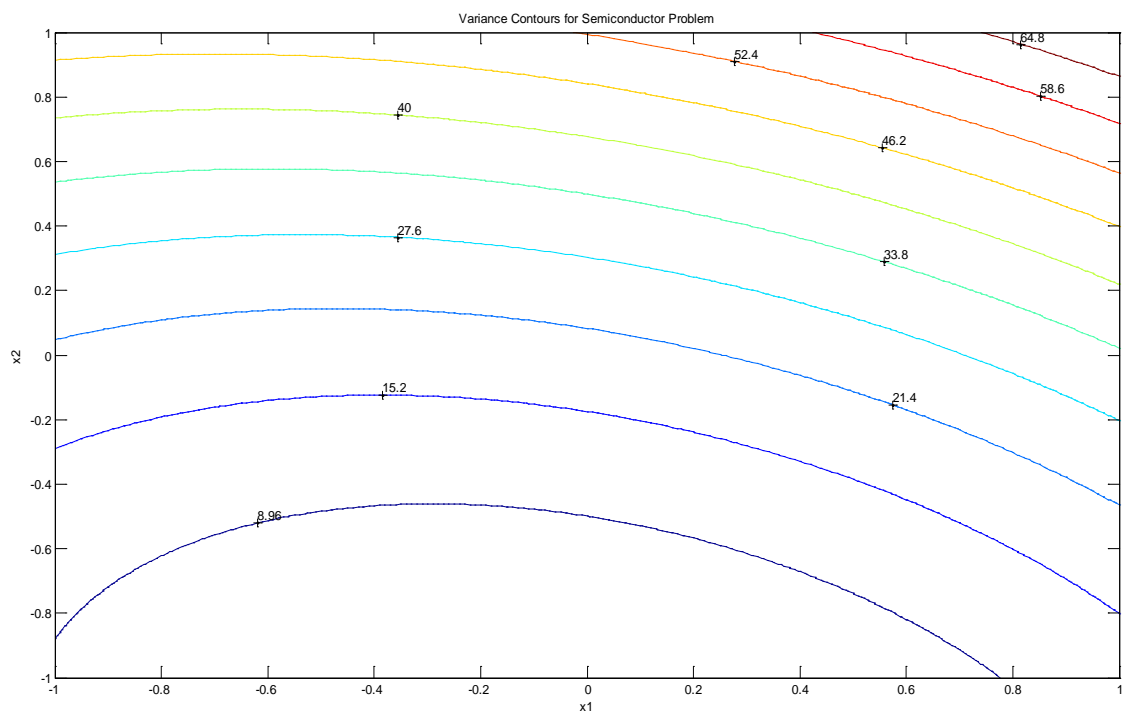


Figure 7. Variance Contour Plot

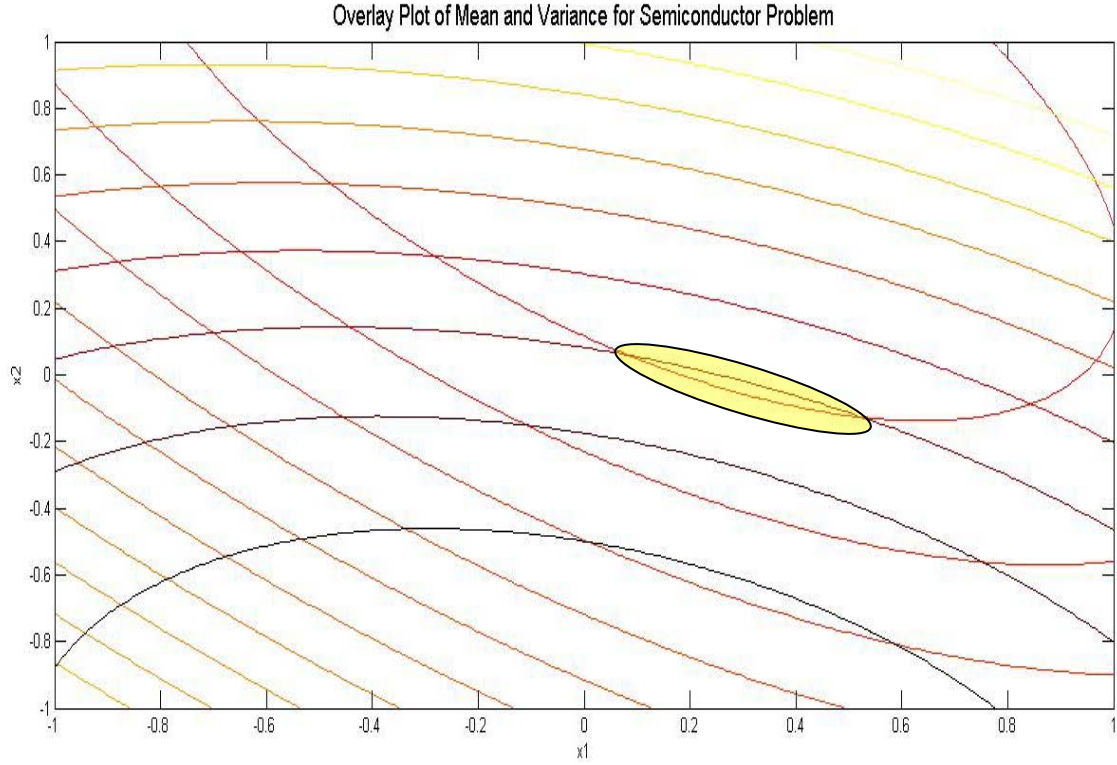


Figure 8. Overlay Contour Plot

The LT solution achieved a smaller mean (28.46) at the cost of a slightly higher variance (standard deviation of 6.5). Both solutions represented “good” solutions by maintaining low variance while satisfying the constraint on the mean value. Multiple approaches that may lead to different optimal solutions based on tradeoffs inherent in their methodologies were presented in Table 1.

This concludes a review on calculating robust optimal control settings using the LT formulation. In the next section, an approach is developed to find control variable settings that guard against future system degradation. A new robust solution will be developed to counteract the possibility of the system degrading quickly.

3.3 Robust Parameter Design and System Degradation

RPD assumes that the process model properly fits the system and that the relationships between the system response, control variables, and noise variables do not change with time. Ideally, after performing RPD, the system would work as predicted. However, as stated by the second law of thermodynamics (Carnot *et al.*, 2005), the quality of matter/energy will deteriorate over time. Hence, it would be desirable to develop modifications to RPD to account for the fact that physical systems will tend to degrade over time, thus reducing the performance of the systems. The performance of software systems can also be degraded by being exposed to inputs beyond the experience of their design and training. If left unconsidered, this inevitable degradation can be an expensive cost. In this research, system degradation is modeled to obtain new settings that continue to remain robust to noise variables (traditional RPD) while becoming robust to changes in the system that diminish performance.

3.3.1 Guarding Against System Degradation

In RPD, the mean performance and the variance of the performance are approximated by low order polynomials. One method of finding a robust solution is to combine these expressions into a composite expression for MSE (LT). This expression becomes an objective function in a minimization problem. This philosophy was presented in Sections 2.2.4 and 3.2.1.

If the system is suffering from performance degradation over time, it seems reasonable to assume that the relationship between the control variables and the system's

mean performance and the variance of that performance will change. This, in turn, suggests that estimated coefficients in the MSE (LT) composite would also change.

Loeffelholz & Bauer (2009) examined this phenomenon by performing an experimental design (DOE) on the coefficients of the mean and variance models which contained two control variables and two noise variables. These authors re-sampled data in the Semiconductor Example problem based on Equation (3.5). A mean and variance model was calculated, similar to those constructed in Equation (3.6) and (3.7). A full two level factorial was performed on the mean and variance models and the results were “crossed” with one another resulting in $64 \times 128 = 8192$ total runs. The high and low coefficient settings (coded as -1 and 1) for each run were calculated as a constant percentage change (δ) in each coefficient for the mean ($\tilde{\beta} = \hat{\beta} \pm \delta(\hat{\beta})$) and variance ($\tilde{\gamma} = \hat{\gamma} \pm \delta(\hat{\gamma})$) models. Each new set of coefficients, $\tilde{\beta}$ and $\tilde{\gamma}$, populated an ensemble of mean models and variance models, respectively. Some of these models exhibited performance improvements, while others exhibited degraded performance.

An LT solution was calculated for all 8192 control variable coefficient combinations. This resulted in LT values that were either higher or lower than the original optimal solution. Only the combination of coefficients that yielded LT values greater (worse) than the original optimal solution were considered as possible candidate coefficient sets that reflected system degradation. The authors chose the coefficient settings which resulted in the worst LT value and obtained new optimal settings that were designed to be more robust to the noise variables in the system as well as certain perturbations in the system causing degradation.

To test these “doubly robust” settings, the system was tested under both normal operating conditions and degraded conditions. Under normal conditions, the original RPD settings outperformed the doubly robust settings, but the difference did not appear significant. When the system was tested under various conditions that caused system degradation, the doubly robust settings outperformed the original RPD settings significantly. This method appeared effective, but the change in coefficients were required to be constant for every term in the model which led to a situation where the actual system degradation, as measured in increase MSE (LT), was unpredictable (Loeffelholz & Bauer, 2009).

There are numerous locations in the coefficient space of the mean and variance models that result in a decrease in performance (increased LT value). A method is desired to guard against conditions where the LT criterion has increased to some preset percentage of its optimum value. One way to find such conditions is to follow the direction of maximum change in the LT function (in terms of the coefficients, given the optimal control settings) until this percentage increase is realized. This is accomplished in the coefficient space by using a simple gradient search, as seen in Figure 9.

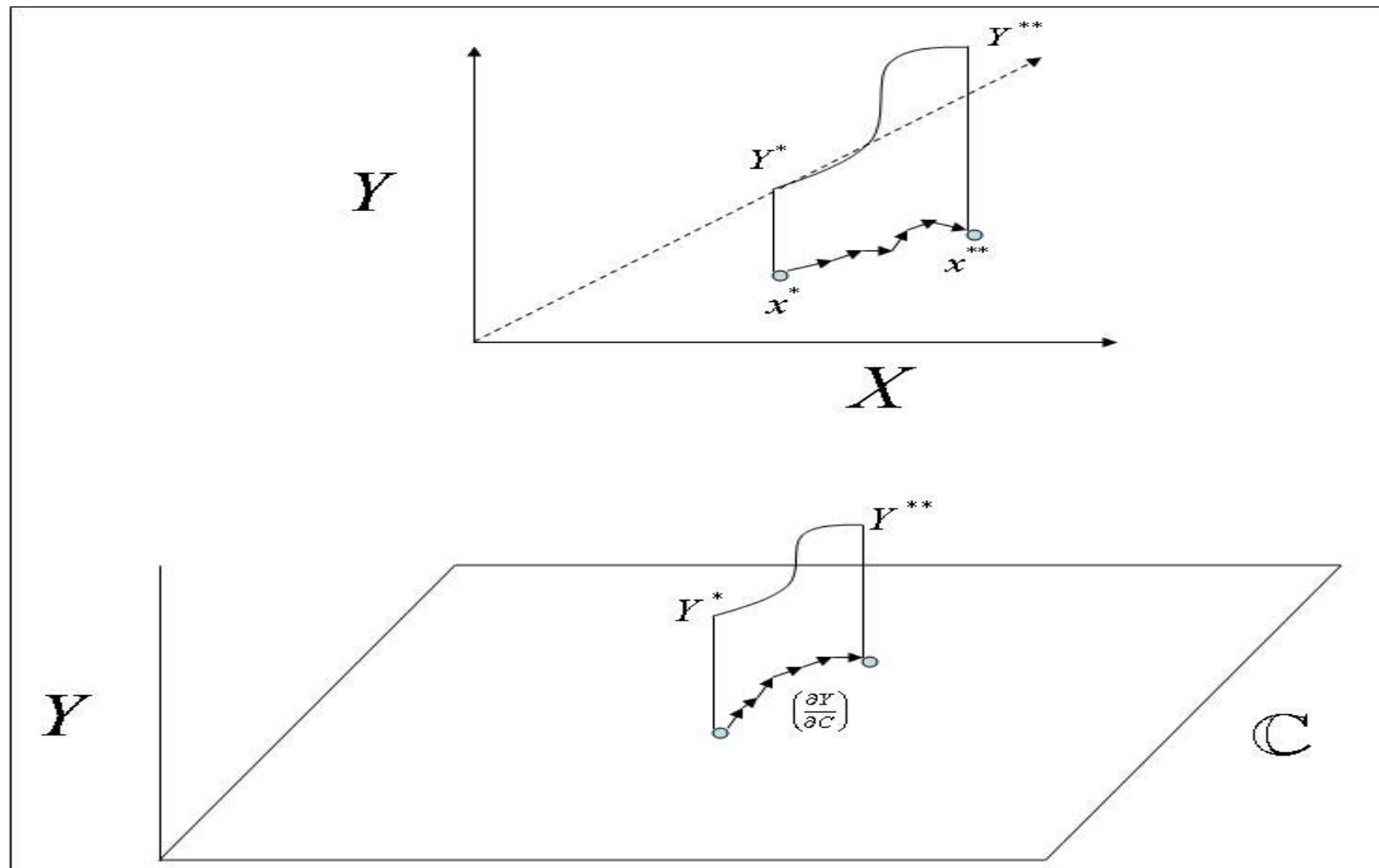


Figure 9. Gradient Search for New Solution

Figure 9 depicts the optimal RPD settings as x^* , which correspond to the optimal LT value (Y^*) at the appropriate coefficients, $C \in \mathbb{C}$, where \mathbb{C} is the coefficient space. The optimal LT value can also be represented as $Y_{(C)}^* = Y(x_{(C)}^*)$. A gradient search, based upon $\left(\frac{\partial Y}{\partial C}\right)$, is performed in \mathbb{C} until a specified percentage increase in LT (Y^{**}) is realized. This percentage increase in LT is calculated as $Y^{**} = \left(1 + \frac{x}{100}\right)Y^*$. Prior to the gradient search, $C^{initial} = C$, and at the culmination of the gradient search, $C^{degrade} = C$. The LT problem is then resolved at this point $(C^{degrade})$ to obtain the doubly robust settings, x^{**} .

For notational purposes, several terms are defined. Given the mean and variance models in Equations (3.2) and (3.3), the general form of the LT (Y) function for a minimization problem is:

$$Y = \{E_{(z,\varepsilon)}[\hat{y}(x, z)]\}^2 + V_{(z,\varepsilon)}[\hat{y}(x, z)] \quad (3.9)$$

Equation 3.9 can be rewritten as (assuming $\sigma_z^2 = 1$):

$$Y = \{\beta_0 + x' \beta + x' Bx\}^2 + (\gamma + \Delta' x)'(\gamma + \Delta' x) + \sigma^2 \quad (3.10)$$

For differentiating purposes, Equation (3.10) can be rewritten as:

$$Y = \Phi + \Psi \quad (3.11)$$

For p control variables and q noise variables, the elements of Equation (3.11) are defined in Equation (3.12) and (3.13). β_{jk} represent the coefficients of the B matrix previously defined as the control by control interaction coefficients. δ_{ji} represent the

coefficients of the Δ matrix previously defined as the control by noise interaction coefficients.

$$\Phi = \left(\beta_0 + \sum_{j=1}^p \beta_j x_j + \sum_{j=1}^p \sum_{k=j}^p x_j \beta_{jk} x_k \right)^2 \quad (3.12)$$

$$\Psi = \sum_{i=1}^q \left(\gamma_i + \sum_{j=1}^p \delta_{ji} x_j \right)^2 \quad (3.13)$$

A gradient search, at small steps, is conducted over the \mathbb{C} space to locate the set of coefficients that lead to a preset percentage degradation in LT performance. Since the gradient is used, the degradation is achieved in the quickest possible fashion. The gradient of each coefficient in the LT function is calculated. In general, differentiating Φ with respect to its coefficients yields:

$$\frac{\partial \Phi}{\partial \beta_0} = \frac{\partial}{\partial \beta_0} \left(\beta_0 + \sum_{j=1}^p \beta_j x_j + \sum_{j=1}^p \sum_{k=j}^p x_j \beta_{jk} x_k \right)^2 = 2 \left(\beta_0 + \sum_{j=1}^p \beta_j x_j + \sum_{j=1}^p \sum_{k=j}^p x_j \beta_{jk} x_k \right) \quad (3.14)$$

$$\frac{\partial \Phi}{\partial \beta_j} = \frac{\partial}{\partial \beta_j} \left(\beta_0 + \sum_{j=1}^p \beta_j x_j + \sum_{j=1}^p \sum_{k=j}^p x_j \beta_{jk} x_k \right)^2 = 2 \left(\beta_0 + \sum_{j=1}^p \beta_j x_j + \sum_{j=1}^p \sum_{k=j}^p x_j \beta_{jk} x_k \right) x_j \quad (3.15)$$

$$\frac{\partial \Phi}{\partial \beta_{jk}} = \frac{\partial}{\partial \beta_{jk}} \left(\beta_0 + \sum_{j=1}^p \beta_j x_j + \sum_{j=1}^p \sum_{k=j}^p x_j \beta_{jk} x_k \right)^2 = 2 \left(\beta_0 + \sum_{j=1}^p \beta_j x_j + \sum_{j=1}^p \sum_{k=j}^p x_j \beta_{jk} x_k \right) x_j x_k \quad (3.16)$$

In general, differentiating Ψ with respect to its coefficients yields:

$$\frac{\partial \Psi}{\partial \gamma_i} = \frac{\partial}{\partial \gamma_i} \sum_{i=1}^q \left(\gamma_i + \sum_{j=1}^p \delta_{ji} x_j \right)^2 = 2 \left(\gamma_i + \sum_{j=1}^p \delta_{ji} x_j \right) \quad (3.17)$$

$$\frac{\partial \Psi}{\partial \delta_{ji}} = \frac{\partial}{\partial \delta_{ji}} \sum_{i=1}^q \left(\gamma_i + \sum_{j=1}^p \delta_{ji} x_j \right)^2 = 2 \left(\gamma_i + \sum_{j=1}^p \delta_{ji} x_j \right) x_j \quad (3.18)$$

Therefore, differentiating Equation (3.10) relative to the original coefficients (C) of Y yields the vector:

$$\nabla Y = \frac{\partial Y}{\partial C} = \left(\frac{\partial Y}{\partial \beta_0}, \dots, \frac{\partial Y}{\partial \beta_{pp}}, \frac{\partial Y}{\partial \gamma_1}, \dots, \frac{\partial Y}{\partial \gamma_q}, \frac{\partial Y}{\partial \delta_{11}}, \dots, \frac{\partial Y}{\partial \delta_{qp}} \right)' \quad (3.19)$$

There are $\frac{p^2 + 3p + 2}{2}$ coefficients due to the mean model and $q + pq$ coefficients due to the variance model. Therefore, the gradient vector, ∇Y contains r partial derivatives:

$$r = \left(\frac{p^2 + 3p + 2}{2} \right) + (q + pq) \quad (3.20)$$

For a small step size ξ and $C^{old} = C$, the gradient search has the form:

$$C^{new} = C^{old} + \xi (\nabla Y) \quad (3.21)$$

After each small step of the gradient search, the LT problem containing the new perturbed coefficients needs to be resolved:

$$x^{new} = \arg \min_{x \in D} Y(C^{new}, x) \quad (3.22)$$

where D is the design space for x . Figure 9 depicts the gradient search in \mathbb{C} space as well as the mirrored sequence of optimal settings in the control variable space. This process is repeated until the preset percentage degradation of LT performance (Y^{**}) is realized. At this point, the LT problem is resolved a final time to determine the final optimal control settings. Figure 9 depicts these doubly robust settings as x^{**} , whereas the original LT optimal settings are x^* . Figure 10 summarizes the algorithm used to find a doubly robust solution.

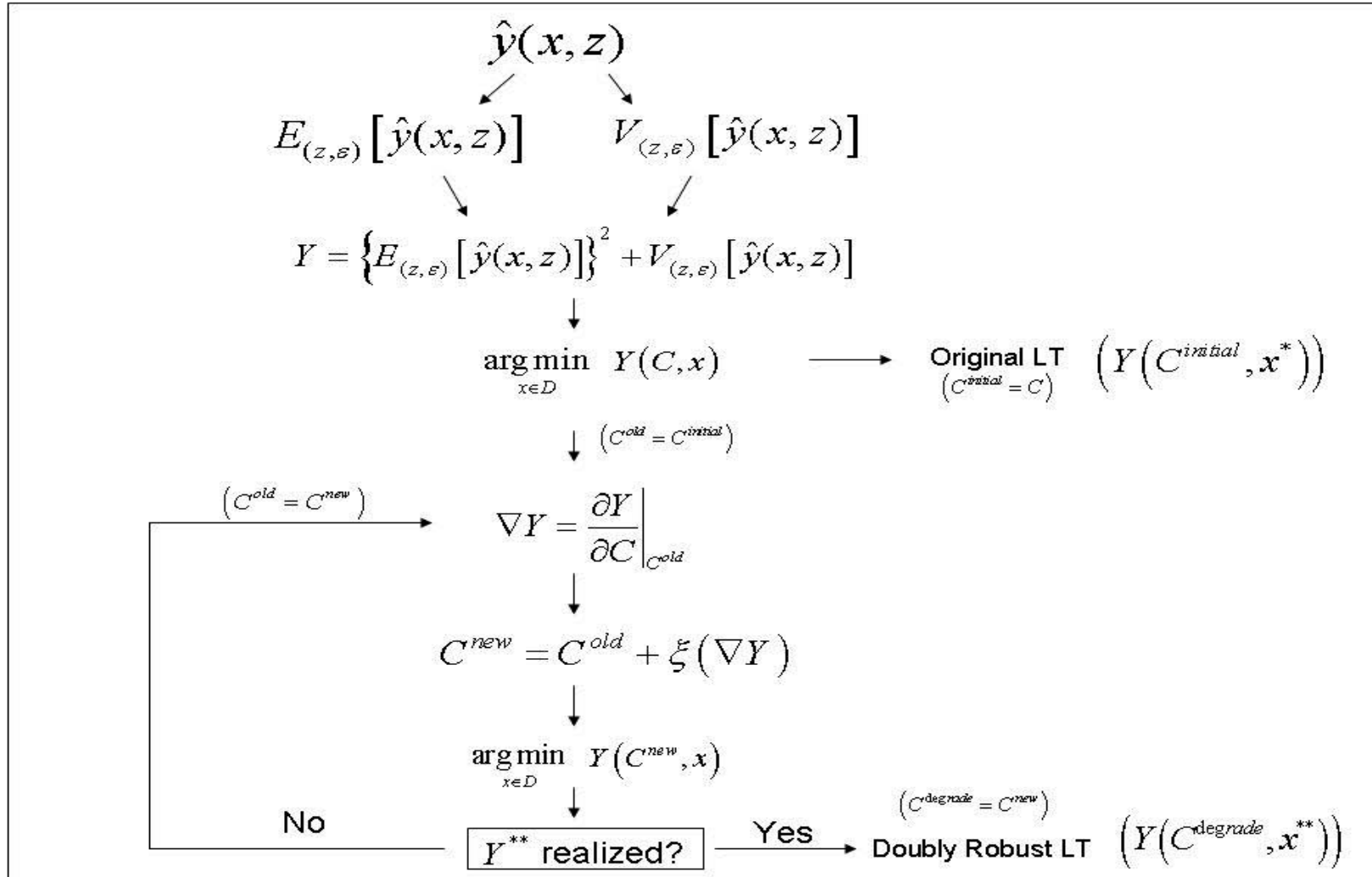


Figure 10. Algorithm for Finding a Doubly Robust Operating Point

3.3.2 Doubly Robust Solution Example

To demonstrate the doubly robust algorithm described in Section 3.3.1, the previous Semiconductor Manufacturing example was implemented. To test the doubly robust solution, a Truth Model was necessary to perform confirmatory trials. Therefore, the process model from Equation (3.5) was used as the Truth Model. For convenience, Equation (3.5) was reproduced as Equation (3.23) below:

$$y(x, z) = 30.37 - 2.92x_1 - 4.13x_2 + 2.6x_1^2 + 2.18x_2^2 + 2.87x_1x_2 + 2.73z_1 - 2.33z_2 + 2.33z_3 - 0.27x_1z_1 + 0.89x_1z_2 + 2.58x_1z_3 + 2.01x_2z_1 - 1.43x_2z_2 + 1.56x_2z_3 + \varepsilon \quad (3.23)$$

The variance of the error term remained 0.9526, in order to obtain a different process model during each re-sampling of $y(x, z)$. The same 23 run CCD was implemented to create new response values. Quadratic regression was performed to obtain the following process model, where $\varepsilon \sim N(0, \sqrt{1.01})$:

$$\hat{y}(x, z) = 29.95 - 3.16x_1 - 4.099x_2 + 2.895x_1^2 + 2.33x_2^2 + 2.72x_1x_2 + 2.67z_1 - 2.497z_2 + 2.38z_3 - 0.37x_1z_1 + 1.06x_1z_2 + 2.79x_1z_3 + 2.04x_2z_1 - 1.38x_2z_2 + 1.85x_2z_3 + \varepsilon \quad (3.24)$$

From $\hat{y}(x, z)$, the mean model and variance models were computed as:

$$E_{(z, \varepsilon)}[\hat{y}(x, z)] = 29.95 - 3.16x_1 - 4.099x_2 + 2.895x_1^2 + 2.33x_2^2 + 2.72x_1x_2 \quad (3.25)$$

$$V_{(z, \varepsilon)}[\hat{y}(x, z)] = (2.67 - 0.37x_1 + 2.04x_2)^2 + (-2.497 + 1.06x_1 - 1.38x_2)^2 + (2.38 + 2.79x_1 + 1.85x_2)^2 + 1.01 \quad (3.26)$$

This problem remained a minimization problem, thus, using Equation (3.9) to calculate LT values for every combination of control settings. This equation was:

$$LT = \left(29.95 - 3.16x_1 - 4.099x_2 + 2.895x_1^2 + 2.33x_2^2 + 2.72x_1x_2 \right)^2 + \left[\begin{aligned} &(2.67 - 0.37x_1 + 2.04x_2)^2 + (-2.497 + 1.06x_1 - 1.38x_2)^2 \\ &+ (2.38 + 2.79x_1 + 1.85x_2)^2 + 1.01 \end{aligned} \right] \quad (3.27)$$

The optimal (minimum) LT value selected yielded control settings [0.22, 0.60].

These settings are similar to the settings presented in Table 4, thus verifying the approach taken with re-sampling data and performing quadratic regression on the model.

A gradient search was then applied to Equation (3.27), where $Y=LT$, to determine doubly robust settings. Equation (3.27) was derived from Equation (3.10) which is given again as Equation (3.28):

$$Y = \left\{ \beta_0 + x' \beta + x' B x \right\}^2 + (\gamma + \Delta' x)' (\gamma + \Delta' x) + \sigma^2 \quad (3.28)$$

For this particular example problem, two control variables and three noise variables were used. Recall the gradient search calculates the partial derivatives for every term in Y with respect to C . A partial derivative is taken for each coefficient in C , thus yielding a single vector of partial derivatives. Note that each partial derivative is a scalar. Therefore, for r terms in C , the general form of the partial derivatives in Equation (3.28) is:

$$\begin{aligned}
\frac{\partial Y}{\partial C} = \frac{\partial Y}{\partial \beta_0} &= 2\{\beta_0 + x' \beta + x' Bx\} \\
\frac{\partial Y}{\partial \beta_1} &= 2\{\beta_0 + x' \beta + x' Bx\} x_1 \\
&\vdots \\
&\vdots \\
\frac{\partial Y}{\partial \beta_p} &= 2\{\beta_0 + x' \beta + x' Bx\} x_p \\
\frac{\partial Y}{\partial \beta_{11}} &= 2\{\beta_0 + x' \beta + x' Bx\} x_1 x_2 \\
&\vdots \\
&\vdots \\
\frac{\partial Y}{\partial \beta_{pp}} &= 2\{\beta_0 + x' \beta + x' Bx\} x_p x_p \\
&\vdots \\
\frac{\partial Y}{\partial \gamma_1} &= 2\{(\gamma_1 + \Delta' x)\} \\
&\vdots \\
&\vdots \\
\frac{\partial Y}{\partial \gamma_q} &= 2\{(\gamma_q + \Delta' x)\} \\
&\vdots \\
\frac{\partial Y}{\partial \delta_{11}} &= 2\{(\gamma_1 + \Delta' x)\} x_1 \\
&\vdots \\
&\vdots \\
\frac{\partial Y}{\partial \delta_{qp}} &= 2\{(\gamma_q + \Delta' x)\} x_p
\end{aligned} \tag{3.29}$$

This particular example contained $r = \frac{2^2 + (3)(2) + 2}{2} + (3 + (2)(3)) = 15$ partial derivatives. After calculating all the partial derivatives, the coefficients in Equation (3.27) were used as the starting point for the gradient search. Small step sizes ($\xi = .001$) were implemented. Following the completion of a single step, the new LT (Y) problem was solved to obtain optimal control settings. As outlined in Figure 10, this process was repeated until $Y^{**} = 1023.28$ was achieved, which is defined as a 20 percent increase in LT. This example ended the gradient search with doubly robust settings, $x^{**} = [0.24, 0.33]$.

A summary of the original LT settings (x^*) and doubly robust settings (x^{**}) is provided in Table 6.

Table 6. Original and Doubly Robust Settings

Settings	x1	x2
Original	0.22	0.6
Doubly Robust	0.24	0.33

The settings were quite different from one another (in terms of coded values), due to the LT contour space in this problem being relatively flat around the original optimal settings. Figure 11 depicts the LT contour plot for this problem across control settings of $[-1, 1]$ for each control variable. The large oval, centered around $[0.22, 0.60]$ indicates the settings within this region contain LT values very close to one another. This provides some rationale as to why the doubly robust solution exhibits good performance under nominal (non-degraded) operating conditions.

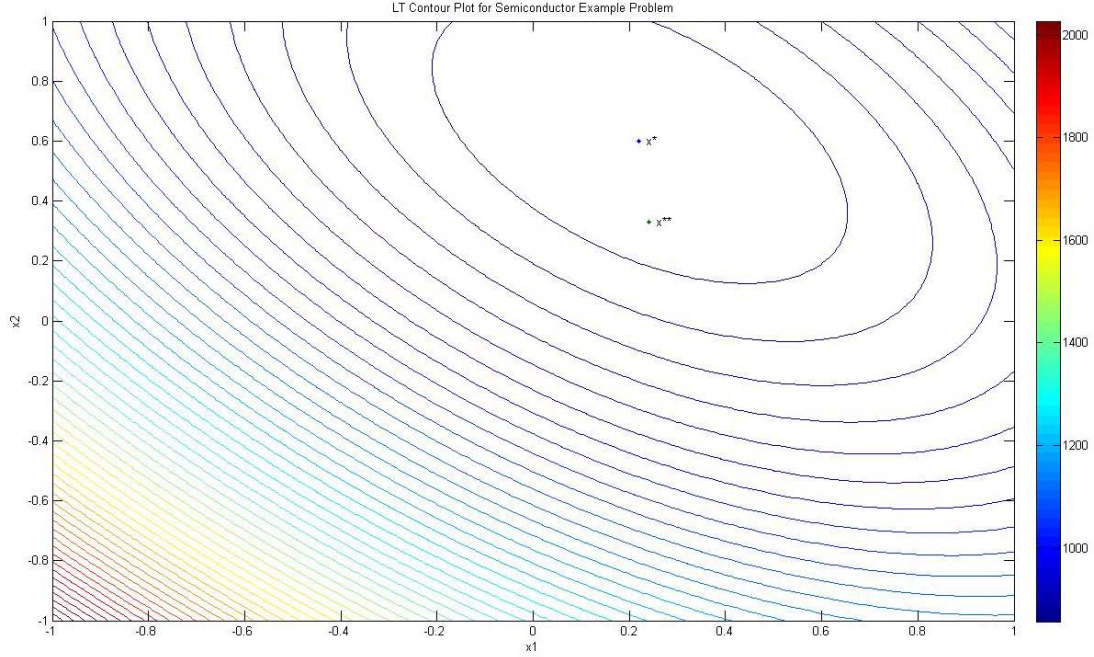


Figure 11. LT Contour Plot for Semiconductor Example Problem

To validate the doubly robust settings, an experiment was conducted in which β_0 and σ_z^2 were varied in Equation (3.27). Varying these terms increases the responses of the mean and variance models respectively, thus causing degraded performance. At normal operating conditions, these values were 29.95 and 1.0 respectively. These values were increased, thus increasing the LT value which in turn displays system degradation. A range of $[0,10]$ was chosen for β_0 and σ_z^2 . For β_0 , the range represented a constant added to the normal operating condition values of the mean response. For σ_z^2 , the range represented a multiplicative effect ($1 + \text{increase}$) on the variance only for values greater than one, thus increasing the variance.

Figure 12 displays the LT contours over the variation of the intercept term and sigma value for the original LT settings, $[0.22, 0.60]$. Figure 13 displays the LT contours

for the doubly robust settings, $[0.24, 0.33]$. As expected, as β_0 and σ_z^2 increased, the LT solution at the given settings increased as well.

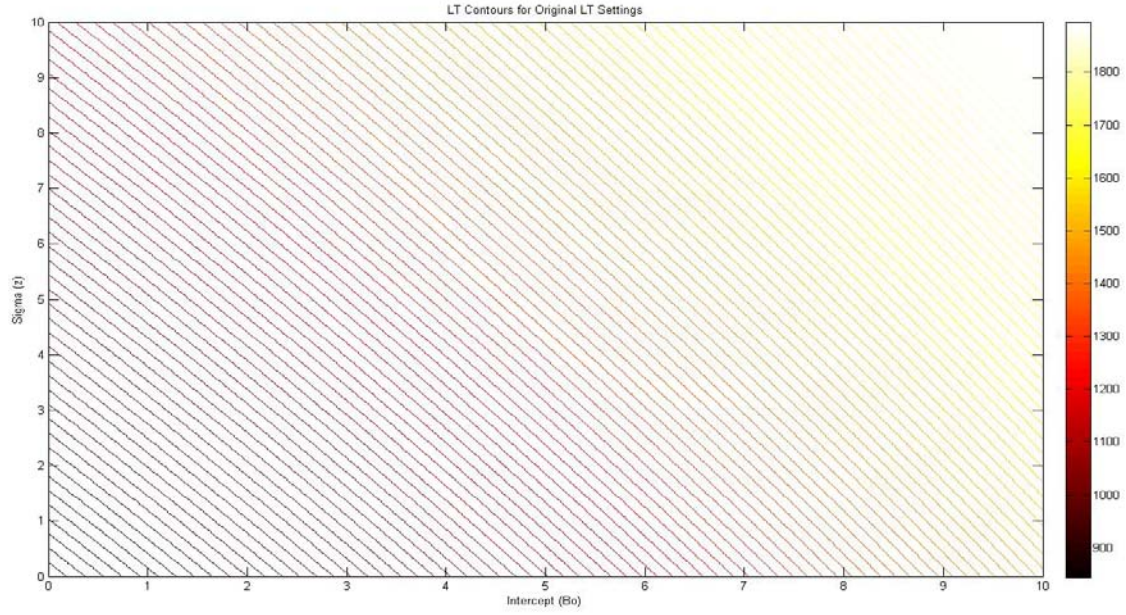


Figure 12. LT Contours for Original LT Settings

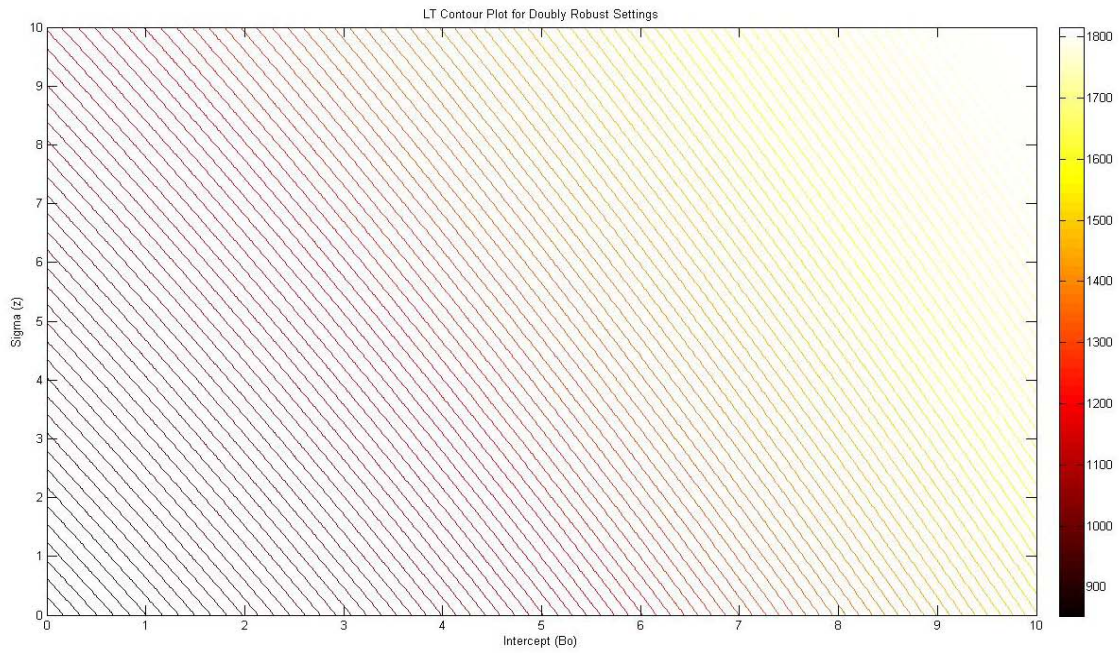


Figure 13. LT Contours for Doubly Robust Settings

To determine the validity of the doubly robust settings, the LT values shown in Figures 12 and 13 were compared. Specifically, the calculation $LT(x^{**}) - LT(x^*)$ was performed. A positive number demonstrated that the Original settings (x^*) outperform the doubly robust settings (x^{**}) and a negative number proposed opposite results. The difference of LT contours between the two sets of settings is given in Figure 14.

The dashed line in the figure represents the boundary which separates regions where the two solutions are preferred. Above the dashed line (positive values) represents the Original settings obtain a lower value and below the dashed line (negative values), doubly robust settings achieve lower LT values. Since the contours change very little when the intercept term was adjusted, the intercept appeared to have little effect on straying from the Original solution. However, once the σ_z^2 value was increased, thus increasing the variance of the system, the doubly robust settings were preferred.

An important note to make is the scale of the contours, located on the z-axis of the figure. Although below the dashed line the Original settings were preferred, the doubly robust settings were not far behind. The largest difference in which the Original settings were preferred is by 13, which occurred by inflating the intercept term well beyond a necessary boundary. However, when increasing the sigma value in small steps, the doubly robust settings drastically moved further away, in terms of LT value, from the Original settings.

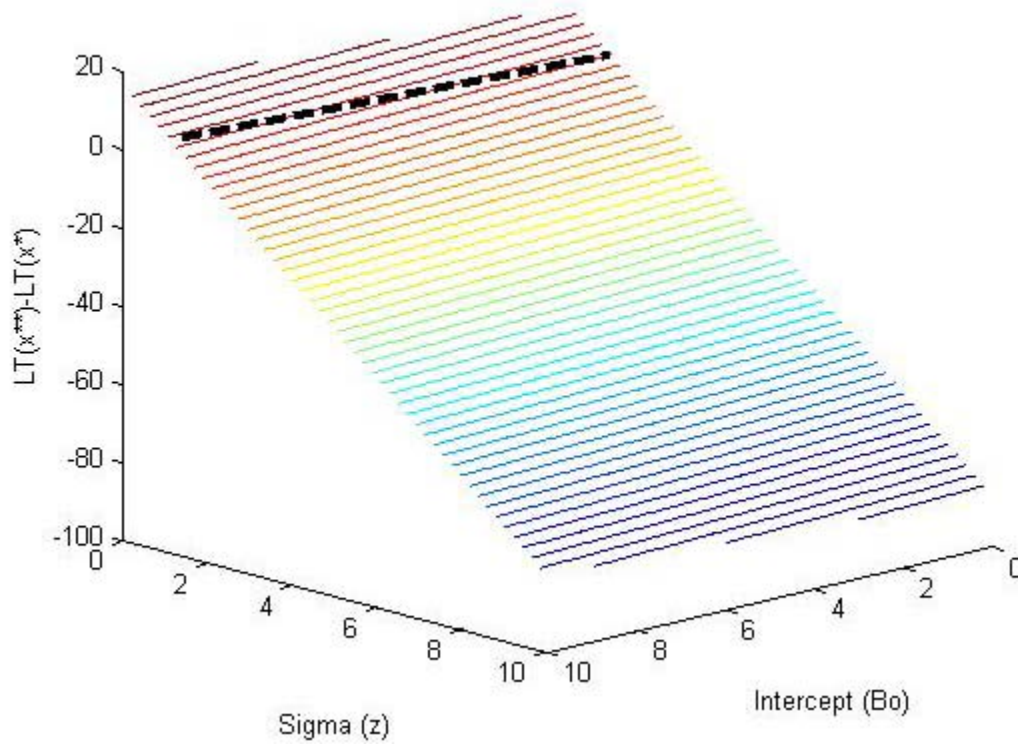


Figure 14. $LT(x^{}) - LT(x^*)$ Contour Plot**

These results verify the algorithm employed to determine doubly robust settings. Not only are the settings robust to noise variables, they are also robust to perturbations in the system causing degradation in performance. The doubly robust settings prove invaluable if a user is uncertain as to whether the system being employed will remain perfectly functional or if over time things may unknowingly occur reducing performance. This research assumes those changes in the system are unknown, because if known, RPD can be re-evaluated to obtain new settings reflecting the changed system.

3.3.4 RPD Summary

Section 3.3 provided methodology on utilizing gradient analysis in a quadratic regression framework to guard against system degradation. In terms of minimal or severe

degradation, the doubly robust settings proved to be more robust than the original LT settings. Also, under normal operating conditions, the doubly robust settings proved competitive.

Much of the literature cited in Chapter 2 addresses different issues in RPD such as solving the dual response problem. However, little in the literature suggests alternative methods for deriving mean and variance models other than the use of quadratic regression. Myers & Montgomery (2002: 562) state “we do not mean to rule out the use of interaction in noise or higher than quadratic terms...however, the model [in Section 2.2.3] will accommodate many real-life situations.” An alternative is explored in Section 3.4 as artificial neural networks are implemented when quadratic regression poorly fits the given data or significant lack of fit is realized.

3.4 Artificial Neural Networks

Some problems may be highly non-linear in nature and cannot be accurately modeled by quadratic regression. In addition, standard RPD methodologies do not model interactions between noise variables which may be important. Properly applied, artificial neural networks (ANNs) allow for the modeling of higher order terms and/or noise variable interactions, thus providing the flexibility to fit non-linear data. Therefore, the use of ANNs to model the process model and/or the mean and variance models appears appropriate.

Radial Basis Function Neural Networks (RBFNNs) and Generalized Regression Neural Networks (GRNNs) were selected for this research. These ANNs are computationally efficient relative to training. However, if quadratic regression can model

the process model accurately, it should be utilized rather than ANNs due to its parsimony and broader level of familiarity to the typical practitioner.

3.4.1 ANNs and RPD

Two approaches were developed to apply ANNs in RPD. In one approach the ANN performs “post-processing” and the other, it performs “pre-processing” of the input data. The former uses control and noise variables as inputs and the latter only inputs control variables. “Pre-processing” ANNs perform quicker but require the use of a crossed array design to collect appropriate variance values. The “post-processing” ANNs work well with either crossed or combined array designs. To aid in demonstration, a notional crossed array design matrix containing two control, x , and two noise variables, $Z = \{z_1, z_2\}$, ($3^2 \times 2^2$) with response, y , was created.

Table 7. Notional Design Matrix

		z1	-1	-1	1	1
		z2	-1	1	-1	1
x1	x2					
-1	-1		47	13	10	30
-1	0		44	93	81	91
-1	1		53	66	37	58
0	-1		93	55	39	96
0	0		68	34	71	28
0	1		26	23	50	66
1	-1		74	23	34	13
1	0		7	3	50	94
1	1		63	28	80	78

The first ANN developed uses a neural network with a single response as the output, where $A = A^{post}$:

$$y = A(x; z) \quad (3.30)$$

The entire dataset in Table 7 was implemented by using the control and noise variables as the inputs. The output response was averaged across all noise variable combinations (N) for each unique combination of control settings. This value represents the expected value for each combination of control settings, as denoted in Equation (3.31):

$$\hat{u}_y(x) = N^{-1} \sum_{z_i \in Z} A(x; z_i) \quad (3.31)$$

Variance values are calculated in the same manner by determining the variance across all noise variable combinations for every control setting as denoted in Equation (3.32):

$$\hat{\sigma}_y^2(x) = \frac{\sum_{z_i \in Z} A(x; z_i)^2 - \left(\sum_{z_i \in Z} A(x; z_i) \right)^2 / N}{(N-1)} \quad (3.32)$$

In this example, the result would be nine control variable combinations, each with an expected value and variance value.

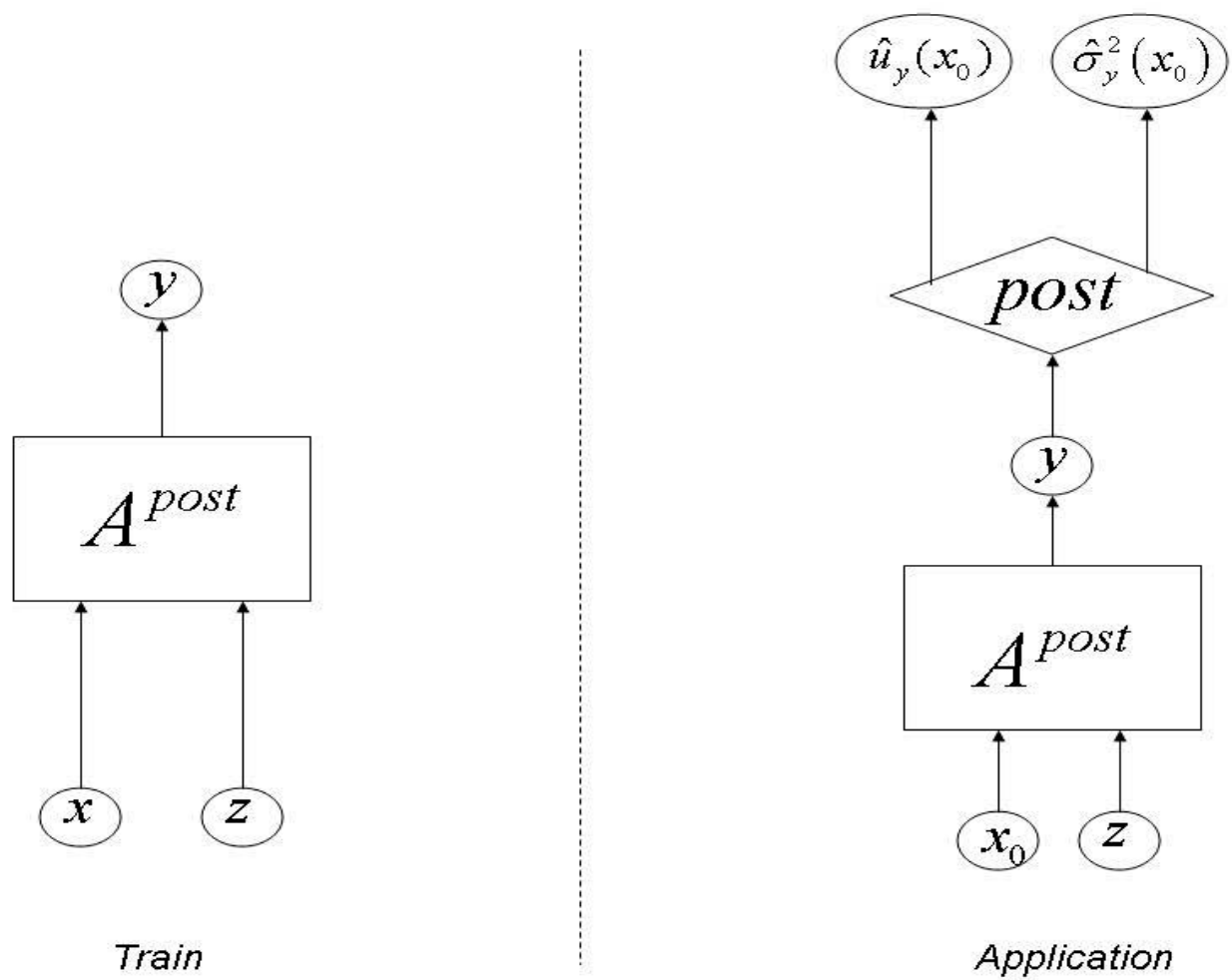


Figure 15. Approach 1 to Develop ANN for RPD

The previous approach requires a large number of inputs into the ANN due to the inclusion of noise variables. To account for this, “pre-processing” can be done to reduce the number of inputs but this approach requires a crossed array design. The next ANN (A^{pre}) constructed takes a single neural network and provides two outputs: mean and variance. Prior to executing the ANN, the mean (Equation (3.31)) and variance (Equation (3.32)) of each row in Table 7 was calculated, which represents the “pre-processing” stage. The unique control settings were used as the inputs of each model with their respective outputs. This notional data is displayed in Table 8.

Table 8. Pre-Processed Data for Mean and Variance

x1	x2	y (mean)	y (variance)
-1	-1	25	300
-1	0	77	520
-1	1	54	153
0	-1	71	800
0	0	50	497
0	1	41	433
1	-1	36	701
1	0	39	1833
1	1	62	578

This network is efficient due to the small number of input variables. However, one drawback is the requirement of sufficient data to accurately model the variance, suggesting the use of a crossed array design; since, if only one combination of noise variable settings is taken for each set of control setting combinations, insufficient data exists for variance estimation. This ANN has two outputs. Alternatively, two networks with one output could also be used.

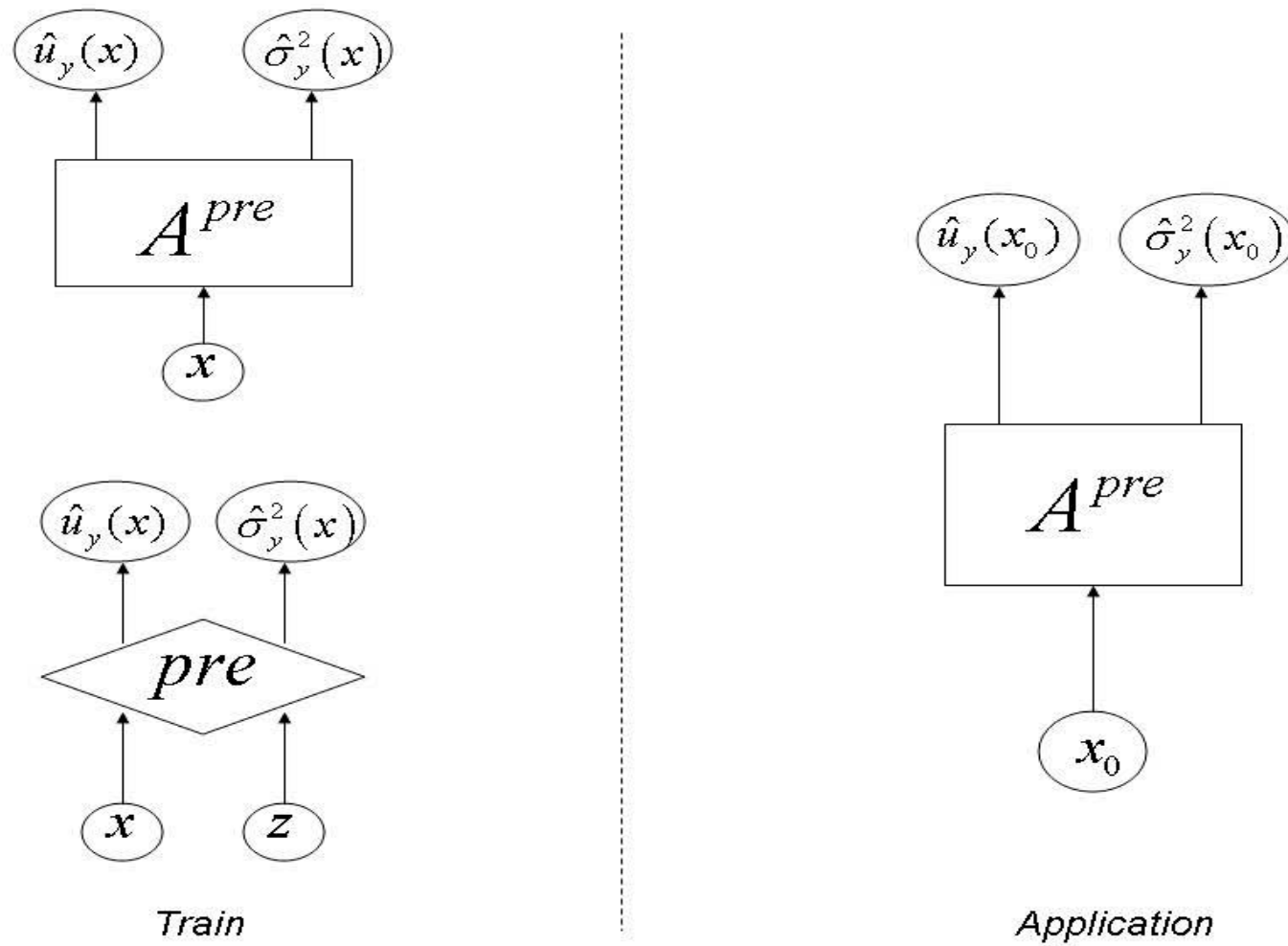


Figure 16. Approach 2 to Develop ANN for RPD

3.4.2 Semiconductor Extended Example Using ANNs in RPD

The example applied in Section 3.2.1 was adjusted to represent a more difficult non-linear problem. Terms were added to the model to include third and fourth order terms as well as interactions between noise variables. The following model was used as the Truth Model:

$$\begin{aligned} y(x, z) = & 30.37 - 2.92x_1 - 4.13x_2 + 2.6x_1^2 + 2.18x_2^2 + 2.73z_1 - 2.33z_2 \\ & + 2.33z_3 - 0.27x_1z_1 + 0.89x_1z_2 + 2.58x_1z_3 + 2.01x_2z_1 - 1.43x_2z_2 + 1.56x_2z_3 \\ & + 3.8094x_1^4 + 2.163x_2^4 + 2.9954x_1^3 + 4.8661x_1^2x_2^2 + 3.4496x_1^2x_2 + 2.0059x_1x_2^3 \\ & - 0.0085z_1^2 + 3.56z_2^2 + 2.8541z_3^2 + 1.3269z_2^4 + 2.624z_1z_2 - 4.5689z_2z_3 + \varepsilon \end{aligned} \quad (3.33)$$

$$\varepsilon = Normal(0, \sqrt{0.9536})$$

To determine the true optimal settings for the Truth Model, an exhaustive search was performed between all possible combinations of control variables and noise variables at a step size of .01, using coded levels. For each combination of control variables settings, the response was averaged (mean) and the variance obtained across all noise variable settings was calculated. These values were then applied to the LT formulation for a minimization problem (Equation (2.12)). This process was replicated 100 times and the optimal LT settings are given in Figure 17. After obtaining 100 control settings and associated LT values, the results were averaged and are reported in Table 9. These optimal settings provide a basis for comparison of results obtained through ANNs versus quadratic regression.

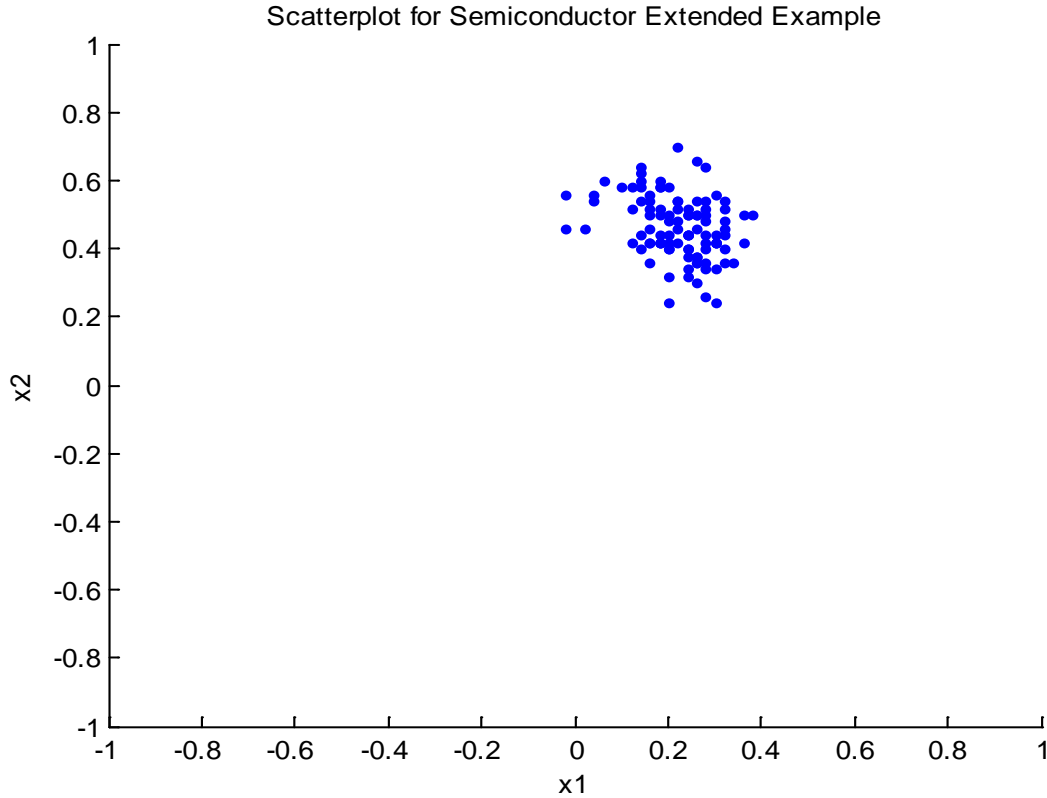


Figure 17. Scatterplot of Optimal LT Settings over 100 Replications

Table 9. True Optimal Settings for Equation (3.27)

x1	x2	Actual LT
0.24	0.48	1382.4

To re-sample data from the Truth Model (Equation 3.33), a $3^2 \times 2^3$ crossed array design was utilized to support both ANN approaches outlined in this research. After obtaining response values for each treatment row in the crossed array design, quadratic regression was performed to fit a process model. The mean and variance models were computed thus allowing for calculation of LT values for all control settings between $[-1,1]$. Quadratic regression yielded optimal control variable settings of $[1, 0.31]$. The actual LT value of these control settings were obtained from the exhaustive search

performed on the Truth Model. These settings corresponded to an LT value of 2108 which is 52.5 percent larger than the LT value of the true optimal settings.

To understand how well quadratic regression performs, Table 10 displays the ANOVA table for this example. The ANOVA table displays a very large p-value, thus making the current quadratic model not significant. The desired p-value is typically denoted as $\alpha = 0.10$ or $\alpha = 0.05$.

Table 10. ANOVA for Semiconductor Extended Example

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F
Model	1525.40	14	108.96	0.77	0.6988 not significant
Residual	8096.91	57	142.05		
Cor Total	9622.31	71			

The coefficient of determination, or R^2 , provides insight into the amount of variability of the data set captured by the model (Montgomery *et al.*, 2004). Along with adjusted R^2 , these value determine how well the model fits the given data. This particular model obtained an $R^2 = 0.1585$, indicating a poor fit.

To demonstrate how poor the quadratic regression fits the given data, since the Truth Model is known, the predicted model can be explored. Table 11 provides the coefficients for the predicted model when sampling data from Equation (3.33), as well as the true coefficients. These estimated coefficients are drastically different than the true coefficients.

Table 11. Coefficients for Semiconductor Extended Example

Factor	Estimate	Truth
Intercept	46.27	30.37
x1	-0.10	-2.92
x2	-1.10	-4.13
z1	-1.19	2.73
z2	2.08	-2.33
z3	0.94	2.33
x1x2	-0.24	0.00
x1z1	1.82	-0.27
x1z2	-2.60	0.89
x1z3	2.18	2.58
x2z1	0.17	2.01
x2z2	0.01	-1.43
x2z3	2.15	1.56
x1^2	-0.11	2.60
x2^2	1.84	2.18
x1^4	0.00	3.81
x2^4	0.00	2.16
x1^3	0.00	2.99
(x1^2)(x1^2)	0.00	4.86
(x1^2)(x1)	0.00	3.45
(x1)(x1^3)	0.00	2.01
z1^2	0.00	-0.01
z2^2	0.00	3.56
z3^2	0.00	2.85
z2^4	0.00	1.32
z1z2	0.00	2.62
z2z3	0.00	-4.57

Along with the ANOVA table and summary statistics, several residual plots were examined. Figure 18 presents the normal probability plot for this example. The plot shows a light tailed distribution on the ends. This could indicate several outliers “pulling” the least squares estimates from their true values.

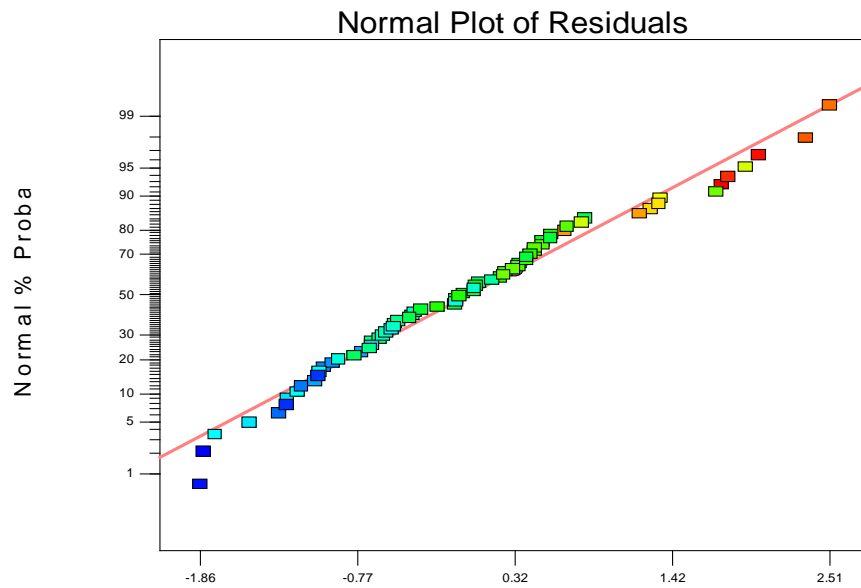


Figure 18. Normal Probability Plot for Semiconductor Extended Example

The plot for predicted values versus actual values is displayed in Figure 19. This plot helps determine the model's predictability, given new observations. This plot allows a visual interpretation of the predicted R^2 value. As seen in Figure 19, this model poorly predicts new observations.

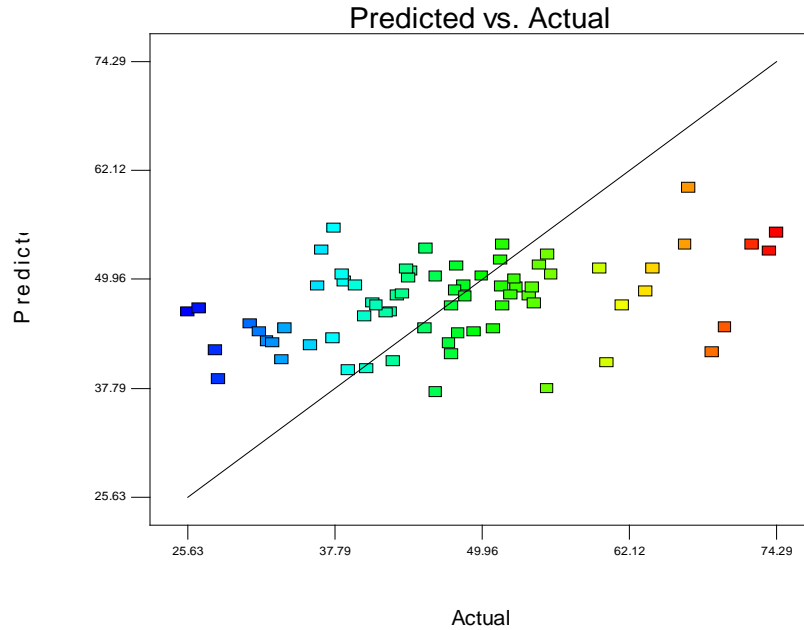


Figure 19. Predicted vs Actual for Semiconductor Extended Example

Analysis in Tables 10-11 and Figures 18-19 indicate that the robust optimal settings obtained through quadratic regression may not truly represent the optimal settings. In fact, according to Table 9 which gives the optimal settings for the Semiconductor Extended example, the quadratic regression settings are nowhere near the optimal. Thus, this analysis indicates that the use of ANNs may be appropriate to better model Equation (3.33).

When developing an ANN, a spread parameter must be defined. Section 2.3.2 and 2.3.3 defined the spread parameter as it is pertinent to RBFNNs and GRNNs. MATLAB® calculates this parameter as $0.8326/\text{spread}$, where spread is user preference (typically a default of 0.1 for RBFNNs and 1.0 for GRNNs). Prior to analysis, a small set of the data was withheld to optimize this value for the optimal spread, which could change based on the holdout dataset. Once the spread was determined for both the

RBFNN and GRNN, the two ANN approaches (1=post-processing , 2= pre-processing) were applied to the full dataset to determine robust settings, assuming a minimum LT existed. Table 12 reports the true optimal and quadratic regression settings as well as the ANN results. As seen in Table 12, the ANNs outperform quadratic regression regardless of the approach or type of neural network used. In fact, the RBFNNs obtain the true optimal in each instance. These results demonstrate the potential of ANNs when quadratic regression fails to properly model the problem.

Table 12. ANN Results on Semiconductor Extended Example

Method		x1	x2	Actual LT
ANN 1	Actual	0.24	0.48	1382
	QR	1	0.31	2108
	RBFNN	0.24	0.48	1382
	GRNN	0.18	0.52	1383
ANN 2	RBFNN	0.24	0.48	1382
	GRNN	0.24	0.48	1382

3.4.3 Koksoy Problem Using ANNs in RPD

The Semiconductor Extended example in Section 3.4.2 used a crossed array design thus providing more data points. This crossed array design used was tailored for ANN approach 2. An example was adapted from Koksoy (2008) that uses a combined array design for three control variables and two noise variables. A CCD was used consisting of 25 runs and response values were provided for two outputs. Y_1 needed to achieve a target value of 1.0 while Y_2 was to be minimized (≥ 0). Table 13 displays the design and responses for this example problem.

Table 13. Example Problem 2 (Koksoy, 2008)

The experimental results for the force transducer experiment

Run	x_1	x_2	x_3	z_1	z_2	y_1	y_2
1	-1	-1	-1	-1	1	1.81	1.10
2	-1	-1	-1	1	-1	1.69	1.11
3	-1	-1	1	-1	-1	1.90	1.07
4	-1	-1	1	1	1	1.78	1.07
5	-1	1	-1	-1	-1	1.80	1.47
6	-1	1	-1	1	1	1.63	1.18
7	-1	1	1	-1	1	1.92	1.41
8	-1	1	1	1	-1	1.78	1.58
9	1	-1	-1	-1	-1	1.36	1.57
10	1	-1	-1	1	1	1.22	2.03
11	1	-1	1	-1	1	1.48	1.38
12	1	-1	1	1	-1	1.44	1.68
13	1	1	-1	-1	1	0.693	3.37
14	1	1	-1	1	-1	0.616	3.75
15	1	1	1	-1	-1	0.950	2.81
16	1	1	1	1	1	0.817	2.83
17	-1	0	0	0	0	1.79	1.24
18	1	0	0	0	0	1.03	2.46
19	0	-1	0	0	0	1.53	1.23
20	0	1	0	0	0	1.22	1.73
21	0	0	-1	0	0	1.30	1.63
22	0	0	1	0	0	1.44	1.67
23	0	0	0	0	0	1.38	1.73
24	0	0	0	0	0	1.39	1.74
25	0	0	0	0	0	1.40	1.74

The two responses, Y_1 and Y_2 , were treated as separate problems simply to demonstrate the ANNs' ability to model problems with combined array designs. Optimal settings were not provided in Koksoy (2008), but can be estimated by examining Table 13. Table 14 provides the estimated optimal robust settings for the two problems.

Table 14. Estimated Optimal Settings for Koksoy Example

Response	x_1	x_2	x_3	Est. Mean
Y_1	1	0	0	1.03
Y_2	-1	-1	1	1.07

Although both ANN methods were performed on Example 2, this problem is more suitable for approach 1. Results are shown from approach 1 which uses a single neural network with one response. The mean and variance is extracted from responses of control settings across noise variables. Quadratic regression was also performed for comparison.

The results obtained from the ANNs (RBFNNs and GRNNs obtained the same results) and quadratic regression for the two responses are displayed in Table 15. RPD was performed to create both a mean model and variance model, thus calculating the LT value, but only the estimated mean is reported in Table 15. The ANN achieved the optimal settings for both problems even though a small combined array design was implemented. Quadratic regression was further from the optimal solution; this was further evidence supporting the use of ANNs even in the presence of small data sets.

Table 15. Settings for ANN and QR for Example 2

Method	x1	x2	x3	Est. Mean
QR-Y1	1	0	-1	0.972
ANN-Y1	1	0	0	1.03
QR-Y2	-1	-1	-1	1.11
ANN-Y2	-1	-1	1	1.07

3.4.4 Doubly Robust Operating Points Using ANNs

Methodology was covered in Sections 3.3.2 and 3.3.3 to guard against system degradation by utilizing a gradient search in the coefficient space. Settings were calculated which were robust against noise variables and robust against perturbations in the system causing performance degradation. These points are called “doubly robust” operating points. In this section, gradient analysis is applied to ANNs when quadratic regression is unsuitable.

The ANNs developed in Section 3.4.1 output a mean and variance value for control settings. For gradient analysis, it is necessary to construct an ANN that outputs LT values based on control settings as inputs. Therefore, an extra step is taken to

transform the results of the ANNs outputting mean/variance values to construct a new ANN (A^{LT}), outputting LT values.

After obtaining the mean ($\hat{u}_y(x)$) and variance ($\hat{\sigma}_y^2(x)$) values, regardless of the ANN approach (*pre/post*) utilized, the LT values are calculated for the control settings. The control settings become the inputs for the new ANN (A^{LT}) with the LT values as the expected response. Gradient analysis is now appropriate.

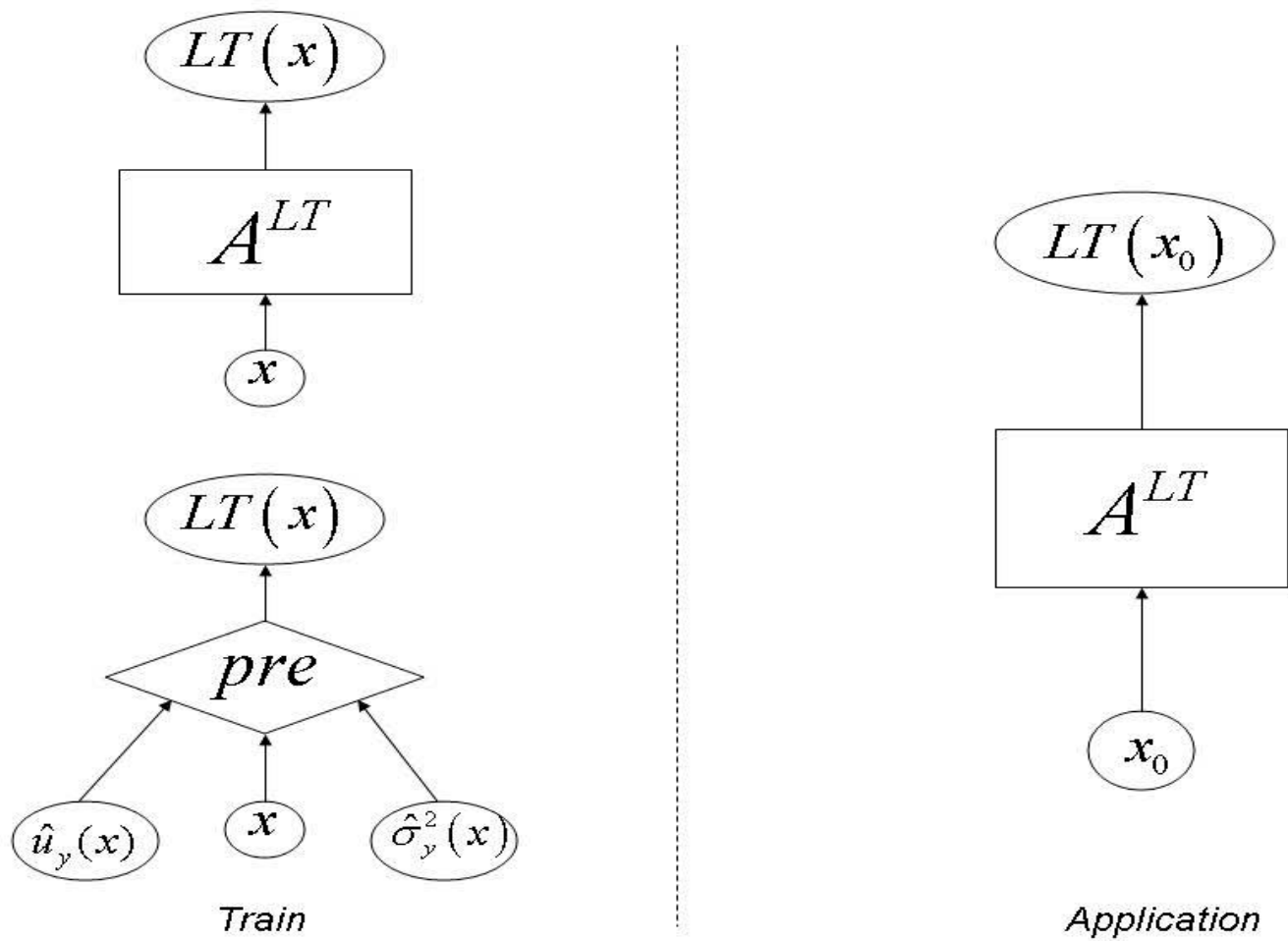


Figure 20. LT ANN for Gradient Analysis

For quadratic regression, the gradient of the LT (Y) function was found with respect to the coefficients of the mean model and variance model. ANNs are parameterized by weights rather than regression coefficients as found in the quadratic models. Figure 20 depicts the ANN to solve for the robust optimal settings (x^*) in the design space (D) of x , under normal operating conditions ($w^{initial}$). This involves solving:

$$x^* = \arg \min_{x \in D} z(w^{initial}, x) \quad (3.34)$$

After solving for the robust optimal settings in Equation (3.34), a gradient search is then performed. To follow the gradient of an ANN, partial derivatives of the output with respect to the weights, $\left(\frac{\partial z}{\partial W}\right)$, need to be calculated. Obtaining these partial derivatives allows one to follow the procedure outlined in Figure 23 where $\nabla z = \left(\frac{\partial z}{\partial W}\right)$. The original weights ($w^{initial}$) are used as the starting point and reassigned as w^{old} . For a small step size, ξ , the gradient search is written as:

$$w^{new} = w^{old} + \xi(\nabla z) \quad (3.35)$$

Following a step in the gradient direction, the optimal control settings are solved:

$$x^{**} = \arg \min_{x \in D} z(w^{new}, x) \quad (3.36)$$

This process is repeated until a preset percent increase in LT (z^{**}) is realized.

The final result yields control settings (x^{**}) that are robust to noise variables and robust to perturbations in the system causing performance degradation.

Herein RBFNNs and GRNNs are examined as neural network options. In the following section, the gradient vector is derived for both the gradient of a RBFNN and then a GRNN.

Recall in Section 2.3.2, the formulation for RBFNNs to calculate an output (z) was:

$$z = \sum_{j=1}^p w_j \exp \left(- \sum_{k=1}^n (b_j^1 (x_k - u_k^{(j)}))^2 \right) + b^2 \quad (3.37)$$

To recap, p centers exist for n features (or input variables). x_k represents the k th input feature/variable of the new exemplar and $\mu_k^{(j)}$ represents the k th component of the j th center. w_j is equal to the weight of the j th node. The initial bias term (b_j^1) represents MATLAB's® interpretation of applying the spread in the equation, which is calculated as $0.8326/\text{spread}$. Also, a second bias (b^2) term is appended to represent a linear layer bias term. Figure 21 represents the RBFNN as outlined in Equation (3.37). The h_p nodes represent each hidden layer node in Equation (3.37).

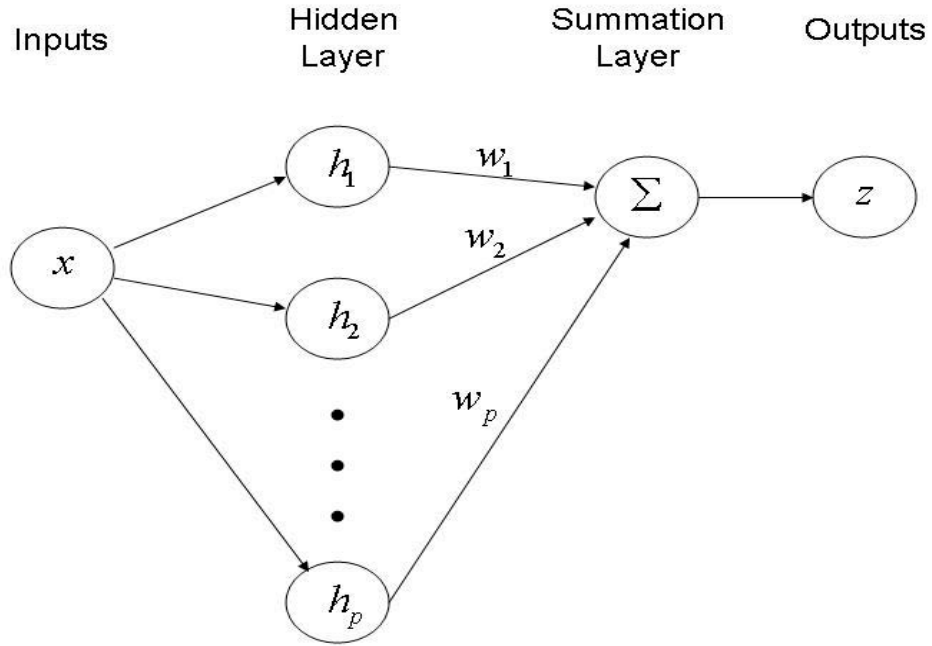


Figure 21. RBFNN with Single Output (LT)

For gradient analysis, the response (z) is the LT value for given input control settings, x . To follow the gradient of an RBFNN, partial derivatives of the output with respect to the weights, $\left(\frac{\partial z}{\partial W}\right)$ where $W = [w_1, w_2, \dots, w_p]$, need to be calculated. This results in a vector of $j=1, 2, \dots, p$ partial derivatives with respect to the output (LT). To calculate the partial derivatives, Equation (3.37) is rewritten as:

$$z = \sum_{j=1}^p w_j c_j + b^2 \quad (3.38)$$

where

$$c_j = \exp\left(-\sum_{k=1}^n (b_j^1 (x_k - u_k^{(j)}))^2\right) \quad (3.39)$$

Equation (3.38) simplifies the problem by assigning c_j and b^2 as constants. Thus, the partial derivative of the output (Equation (3.38)) with respect to the weights reduces to:

$$\frac{\partial z}{\partial W} = [c_1, c_2, \dots, c_p] \quad (3.40)$$

Similar analysis is applied to GRNNs. Recall in Section 2.3.3, the formulation to calculate an output (z) for a GRNN in MATLAB® was:

$$z = \frac{\sum_{j=1}^p w_j \exp\left(-\sum_{k=1}^n (b_j^1(x_k - u_k^{(j)}))^2\right)}{\sum_{j=1}^p \exp\left(-\sum_{k=1}^n (b_j^1(x_k - u_k^{(j)}))^2\right)} \quad (3.41)$$

Notation remains similar to the RBFNN where for n features/variables exist for the input x with its output as z . The initial bias term is calculated the same as in the RBFNN. Figure 22 represents the GRNN outlined in Equation (3.41), where α denotes the numerator and β denotes the denominator.

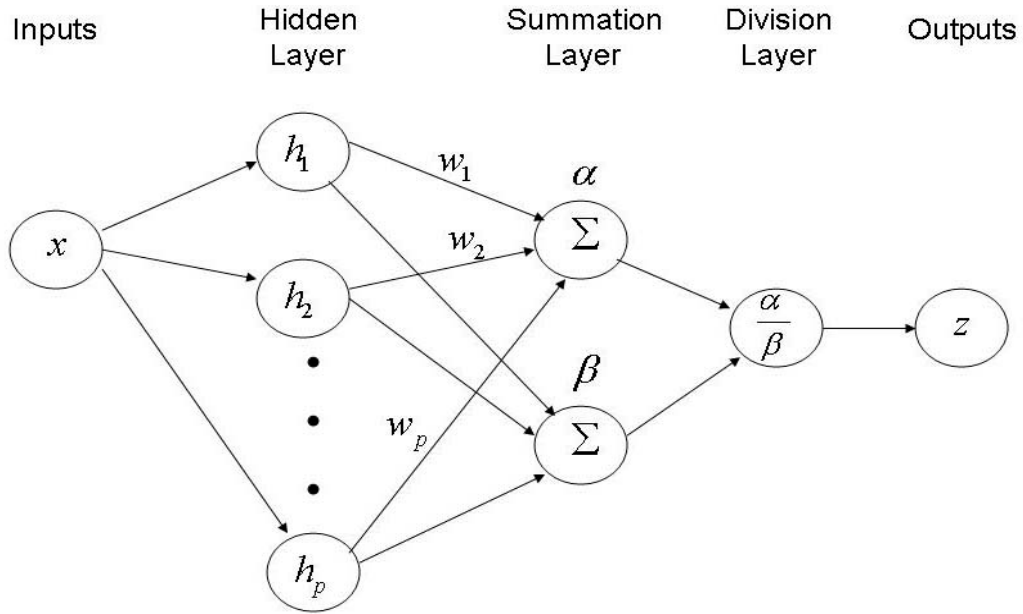


Figure 22. GRNN with Single Output (LT)

For gradient analysis, Equation (3.41) can be rewritten as:

$$z = \frac{\sum_{j=1}^p w_j c_j}{\sum_{j=1}^p c_j} \quad (3.42)$$

where

$$c_j = \exp\left(-\sum_{k=1}^n (b_j^1(x_k - u_k^{(j)}))^2\right) \quad (3.43)$$

The partial derivative of z with respect to the weights, $\left(\frac{\partial z}{\partial W}\right)$, is calculated as:

$$\frac{\partial z}{\partial W} = \left[\frac{c_1}{\sum_{j=1}^p c_j}, \frac{c_2}{\sum_{j=1}^p c_j}, \dots, \frac{c_p}{\sum_{j=1}^p c_j} \right] \quad (3.44)$$

After obtaining the partial derivatives for RBFNNs, Equation (3.40), and GRNNs, Equation (3.44), gradient analysis can be conducted as outlined in Figure 23 and Equations (3.35)-(3.36).

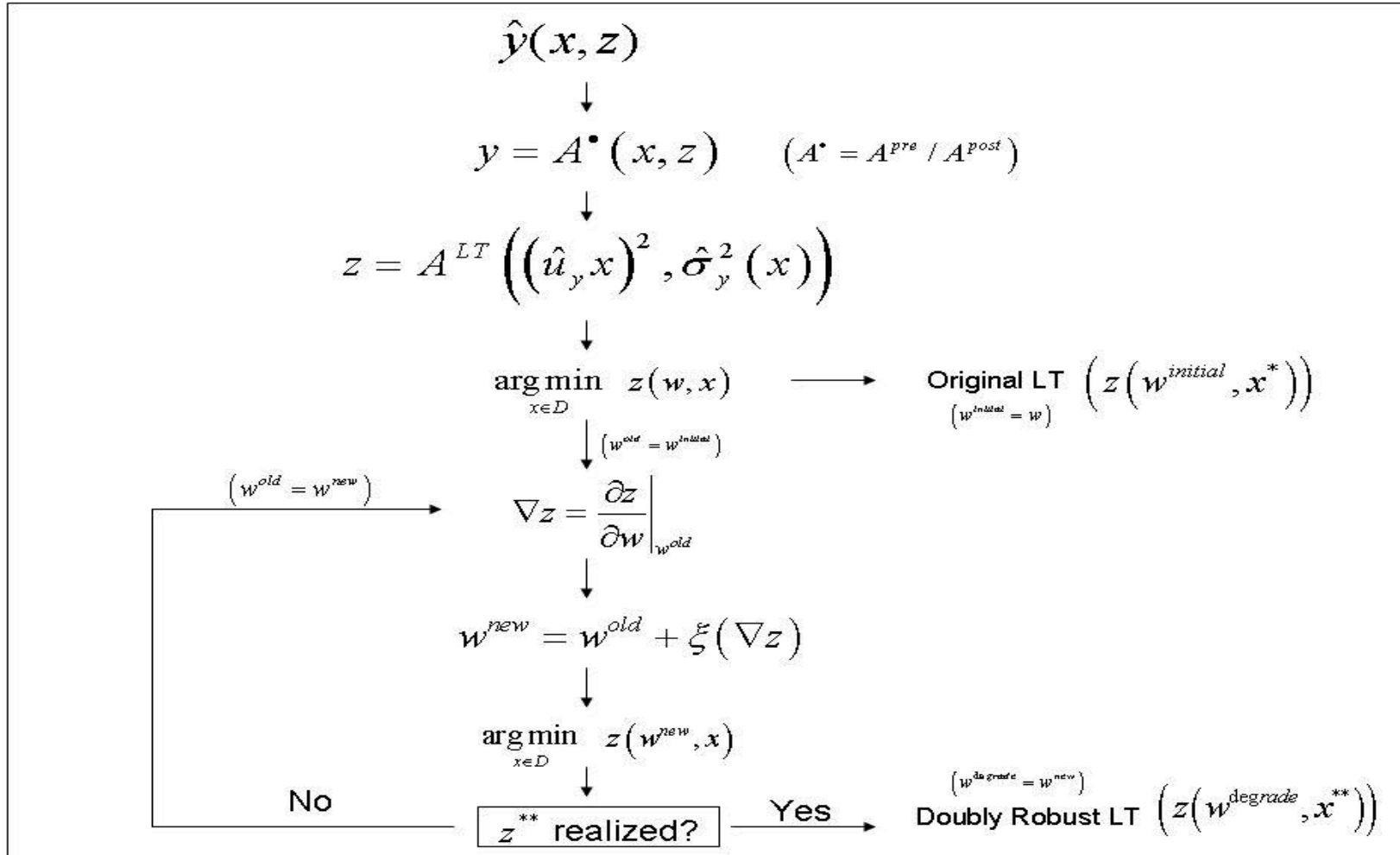


Figure 23. Algorithm for System Degradation in ANNs

Recall that in the Semiconductor Extended example in Section 3.4.2, the system response was defined as:

$$\begin{aligned}
y(x, z) = & 30.37 - 2.92x_1 - 4.13x_2 + 2.6x_1^2 + 2.18x_2^2 + 2.73z_1 - 2.33z_2 \\
& + 2.33z_3 - 0.27x_1z_1 + 0.89x_1z_2 + 2.58x_1z_3 + 2.01x_2z_1 - 1.43x_2z_2 + 1.56x_2z_3 \\
& + 3.8094x_1^4 + 2.163x_2^4 + 2.9954x_1^3 + 4.8661x_1^2x_2^2 + 3.4496x_1^2x_2 + 2.0059x_1x_2^3 \\
& - 0.0085z_1^2 + 3.56z_2^2 + 2.8541z_3^2 + 1.3269z_2^4 + 2.624z_1z_2 - 4.5689z_2z_3 + \varepsilon
\end{aligned} \tag{3.45}$$

$$\varepsilon = \text{Normal}\left(0, \sqrt{0.9536}\right)$$

The optimal settings obtained for this problem using RBFNNs and GRNNs was $[0.24, 0.48]$ with an associated LT value of 1382.4. The weights associated with these ANNs were extracted and implemented as the starting point $(w^{initial})$ for the gradient search. Steps of size .01 in the gradient direction were taken until a 20 percent degradation in LT (z^{**}) was realized, which computes as 1659.

Following the algorithm outlined in Figure 23, doubly robust settings (x^{**}) were calculated. These settings corresponded to $[0.18, 0.58]$ for the RBFNNs and $[0.20, 0.58]$ for the GRNNs. Since a fourth order model was utilized, it was difficult to test the doubly robust conditions under system degradation. For the quadratic regression, one was able to adjust the intercept of the mean model and multiply a constant to the variance. However, literature is scarce as to how to obtain mean and variance models for situations greater than quadratic.

Therefore, these settings were tested against the original settings under normal operating conditions (Equation (3.45)). The solutions for the original settings and their actual LT values are reported in Table 12. This table is extended to include the doubly

robust settings and their actual LT values under normal operating conditions. This is displayed in Table 16.

Table 16. ANN Doubly Robust Settings under Normal Operating Conditions

Method		x1	x2	Actual LT
ANN 1	Actual	0.24	0.48	1382
	QR	1	0.31	2108
	RBFNN	0.24	0.48	1382
	GRNN	0.18	0.52	1383
ANN 2	RBFNN	0.24	0.48	1382
	GRNN	0.24	0.48	1382
Doubly Robust	RBFNN	0.18	0.58	1386
	GRNN	0.2	0.58	1390

The LT values of the doubly robust settings for RBFNNs and GRNNs computed to only a 0.29 and 0.33 percent increase, respectively, in expected LT value from the original settings under normal system operation. This result indicated that selecting doubly robust settings maintained near optimal results under normal conditions. As seen with system degradation in quadratic regression, these settings should have been more robust to system degradation, within the specified LT increase bound. ANNs proved useful in fitting non-linear problems and their use has been adapted to guard against system degradation.

3.5 Multiple Responses

Real world problems often involve measuring multiple responses. Difficulties arise when the optimal choice of settings for each response are different. For example, the Koksoy (2008) example in Section 3.4.3 involved two responses and the optimal settings for each response were on different ends of the spectrum in terms of settings.

This issue becomes increasingly difficult given more response variables. Section 2.4 discussed approaches to multiple response problems, but most involve subject matter expertise for weighting or subjective decision making based on contour plots. To account for this problem, factor analysis (FA) is implemented to discover if the responses can be projected into a meaningful subspace.

3.5.1 Factor Analysis for Multiple Responses

Traditionally, factor analysis is performed on features to determine commonalities for feature reduction. In this research, the same idea is applied to the responses rather than the features (input variables). Applying factor analysis reduces the number of responses to reflect common factors (responses). The factor scores generated represent the new response variable(s). Furthermore, reduction to a single factor allows quadratic regression or ANNs to be applied to find the optimal settings for control variables.

However, factor analysis may only reduce the problem to two or more factors rather than a single factor. The number of responses may have been reduced, but the issue of multiple responses remains. To combat this problem, linear combination techniques are employed to combine factor scores from multiple dimensions into a single dimension. Rotated factor scores were also created using the MATLAB® function *rotatefactors*.

The simplest method to combine the multiple (rotated) factor scores is by addition/subtraction. Signs are attached to the factor scores appropriately to minimize the overall response. For example, if a high factor score is desirable for a particular factor, a negative sign is placed on the factor. Equation (3.46) summarizes this technique for n

factors, where f represents the factor score for factor i and $f^{(R)}$ represents the rotated factor score for rotated factor i as:

$$Y = \sum_{i=1}^n f_i \quad \text{or} \quad Y = \sum_{i=1}^n f_i^{(R)} \quad (3.46)$$

The second linear combination technique applies weights to the (rotated) factor scores. As opposed to subjective weights, eigenvalues contain information on factor importance. The eigenvalues are normalized to represent a value between 0 and 1. Each (rotated) factor is then multiplied by its normalized eigenvalue to produce weighted (rotated) factor scores. Equation (3.47) depicts this method where λ_i represents the weight (normalized eigenvalue) for factor i as:

$$Y = \sum_{i=1}^n \lambda_i f_i \quad \text{or} \quad Y = \sum_{i=1}^n \lambda_i f_i^{(R)} \quad (3.47)$$

The third method considers adjusting the (rotated) factor scores to reflect the same scale. To achieve this, the (rotated) factor scores are normalized. Equation (3.48) outlines methodology to normalize a particular (rotated) factor score, j , within factor i as:

$$\frac{f_{ij} - \min(f_i)}{\max(f_i) - \min(f_i)} \quad (3.48)$$

This normalized score is then added/subtracted similarly to method one (Equation (3.46)). The % symbol represents the normalized (rotated) factor score:

$$Y = \sum_{i=1}^n f_i^{\%} \quad \text{or} \quad Y = \sum_{i=1}^n f_i^{(R\%)} \quad (3.49)$$

The final method combines methods two and three into a single statistic. Normalized (rotated) factor scores are calculated and these new scores are weighted by their normalized eigenvalues, as shown in Equation (3.50):

$$Y = \sum_{i=1}^n \lambda_i f_i^{\%} \quad \text{or} \quad Y = \sum_{i=1}^n \lambda_i f_i^{(R^{\%})} \quad (3.50)$$

Table 17 summarizes the methods created using factor analysis. These methods are only applied if factor analysis reduces the problem to two or more factors. If factor analysis suggests the use of a single factor, these linear combination methods become unnecessary. This single dimension problem permits the use of quadratic regression or ANNs to compute optimal robust settings or doubly robust settings.

Table 17. Summary of (Rotated) Factor Score Reductions

Sum Factor Scores	Sums factor scores appropriately	$Y = \sum_{i=1}^n f_i$
Sum Weighted Factor Scores	Weight factors by the eigenvalues (normalized to 1) and sum appropriately	$Y = \sum_{i=1}^n \lambda_i f_i$
Sum Normalized Factor Scores	Normalize factor score and sum appropriately	$Y = \sum_{i=1}^n f_i^{\%}$
Sum Weighted Norm. Factor Scores	Normalize factor scores, weight factors by eigenvalue, and sum appropriately	$Y = \sum_{i=1}^n \lambda_i f_i^{\%}$
Sum Rot. Factor Scores	Sums rotated factor scores appropriately	$Y = \sum_{i=1}^n f_i^{(R)}$
Sum Weighted Rot. Factor Scores	Weight rotated factor scores by the eigenvalues (normalized) and sum appropriately	$Y = \sum_{i=1}^n \lambda_i f_i^{(R)}$
Sum Normalized Rot. Factor Scores	Normalize rotated factor scores and sum appropriately	$Y = \sum_{i=1}^n f_i^{(R\%)}$
Sum Weighted Norm. Rot. Factor Scores	Normalize rotated factor scores, weight rotated factors by eigenvalue, and sum appropriately	$Y = \sum_{i=1}^n \lambda_i f_i^{(R\%)}$

3.5.2 Factor Analysis Problem

To demonstrate factor analysis on a multiple response problem, a simple five response problem was created. The five responses, y_1, y_2, y_3, y_4, y_5 , required the first three minimized and last two maximized. The problem was constructed to reflect different optimal solutions among several responses. Two control variables and two noise variables were utilized. To re-sample data for each response, a $3^2 \times 2^2$ crossed array design, resulting in 36 runs, was implemented. An error term of $\varepsilon \sim Normal(0, \sqrt{0.9})$ was applied to each model to allow for variation in re-sampling of data. Equations (3.51)-(3.55) represent the true regression models for the five responses.

$$y_1(x, z) = 4.2 + 1.21x_1 - 0.92x_2 - 0.05z_1 + 0.07z_2 + 1.11x_1^2 + 0.88x_2^2 + 1.97x_1x_2 + 0.35x_1z_1 - 0.54x_1z_2 - 0.22x_2z_1 - 0.85x_2z_2 + \varepsilon \quad (3.51)$$

$$y_2(x, z) = 4 + 1.2x_1 - 1.00x_2 + 0.05z_1 + 0.1z_2 + 1.00x_1^2 + 1.00x_2^2 + 2x_1x_2 + 0.55x_1z_1 - 0.65x_1z_2 - 0.18x_2z_1 - 0.90x_2z_2 + \varepsilon \quad (3.52)$$

$$y_3(x, z) = 20 + 2.4x_1 - 2.1x_2 + 1.00z_1 + 0.79z_2 + 2.79x_1^2 - 1.66x_2^2 + 0.94x_1x_2 + 1.00x_1z_1 + 0.01x_1z_2 - 1.31x_2z_1 + 0.11x_2z_2 + \varepsilon \quad (3.53)$$

$$y_4(x, z) = 21 - 4.2x_1 - 2x_2 - 0.60z_1 + 0.90z_2 + 1.1x_1^2 + 0.90x_2^2 + 0.02x_1x_2 - 0.06x_1z_2 + 0.11x_2z_1 + 0.15x_2z_2 + \varepsilon \quad (3.54)$$

$$y_5(x, z) = 19 - 3.9x_1 - 2x_2 + 1.21z_1 - 1.57z_2 - 1.22x_1^2 + 1.00x_2^2 + 2.03x_1x_2 - 0.88x_1z_1 + 0.84x_1z_2 - 1.2x_2z_1 + \varepsilon \quad (3.55)$$

The true optimal robust settings for each response were calculated by using a full five level (5^4) factorial design. This design was replicated 100 times to collect a generous amount of data for accuracy. A mean and variance value was calculated across

all noise variable combinations for each unique control setting combination. Finally, an LT value was computed based on the response being a *minimum* or *maximum*. The minimum LT value was selected in all instances, with the corresponding control settings representing the optimal robust settings. The true optimal settings for each response problem is given in Table 18.

Table 18. Optimal Settings for Five Response Problems

Response	x1	x2	Optimal LT
Y1	-1	1	5.9
Y2	-1	1	4.33
Y3	-0.5	1	237.18
Y4	-1	-1	-855.48
Y5	-1	-1	-695.8

Individually, the settings for the first three responses were similar. Also, the final two responses possessed identical optimal robust settings. Quadratic regression and ANNs were applied to the five responses individually to demonstrate their capability in accurately modeling the problems. As a side note, RBFNNs and GRNNs obtained similar results. Table 19 displays the results, which show accurate modeling was performed using either technique. This was expected for quadratic regression because the true models are quadratic in nature.

Table 19. Settings from QR and ANN for Five Responses

Response	Quadratic Regression		ANNs	
	x1	x2	x1	x2
Y1	-1	1	-1	1
Y2	-1	1	-1	1
Y3	-0.5	1	-0.5	1
Y4	-1	-1	-1	-1
Y5	-1	-1	-1	-1

Factor analysis was applied to the five responses. A factor loadings matrix was constructed to determine which responses could be grouped together. Table 20 displays

the factor loadings matrix which shows two important factors. The first factor grouped the minimized responses while the second grouped the maximized responses. These two factors explained 92% of the variance.

Table 20. Factor Loadings Matrix

	Factor 1	Factor 2
Y1	0.9224	0.3041
Y2	0.9179	0.3205
Y3	0.8017	0.0741
Y4	-0.4531	0.7675
Y5	-0.3422	0.8367

The MATLAB® function, *rotatefactors*, was applied to this data to determine the rotated factors loadings matrix. Three different factors were suggested when rotating the factors. The first two responses was Factor 1, the maximized responses were Factor 2, and the third minimized response corresponded to its own factor. Table 21 displays the rotated factors loadings matrix and the corresponding responses.

Table 21. Rotated Factors Loadings Matrix

	Rot. Factor 1	Rot. Factor 2	Rot. Factor 3
Y1	0.9613	-0.0691	-0.2503
Y2	0.9601	-0.0514	-0.2561
Y3	0.4202	-0.1395	-0.8861
Y4	-0.1767	0.8968	-0.0113
Y5	0.0645	0.8868	0.1803

Factor analysis reduced the five responses into a more manageable two or three response problem depending on whether rotation was utilized. The issue of multiple responses remained evident. Different techniques to handle problems with more than one factor were give in Table 17. All eight techniques were applied for comparison.

Quadratic regression and ANNs were applied to the (rotated) factor scores for modeling. This approach differed from examining factor plots to choose optimal settings.

For simplicity purposes, a step size of .5 was utilized on each control variable. The results are summarized in Table 22, including “General” which simply added/subtracted the standardized response data (Davis, 2009). The table reports the factor analysis method performed, its mathematical notation, and the optimal settings obtained through quadratic regression and ANNs.

Table 22. Robust Solutions for (Rotated) Factor Reduced Methods

Method	Math	Quad. Reg.		ANNs	
		X1	X2	X1	X2
General	$R_1 + R_2 + R_3 - R_4 - R_5$	0.5	0.5	0.5	0.5
Sum Factors	$F_1 - F_2$	-1	-1	-1	-1
Sum Weighted Factors	$\lambda_1 F_1 - \lambda_2 F_2$	-1	1	-1	1
Sum Norm. Factors	$F_1^{\%} - F_2^{\%}$	-1	-1	-1	-1
Sum Weighted Norm. Factors	$\lambda_1 F_1^{\%} - \lambda_2 F_2^{\%}$	-1	1	-1	1
Sum Rot. Factors	$F_1^{(R)} - F_2^{(R)} - F_3^{(R)}$	-1	1	-1	1
Sum Weighted Rot. Factors	$\lambda_1 F_1^{(R)} - \lambda_2 F_2^{(R)} - \lambda_3 F_3^{(R)}$	-1	1	-1	1
Sum Norm. Rot. Factors	$F_1^{(R\%)} - F_2^{(R\%)} - F_3^{(R\%)}$	-1	1	-1	1
Sum Weighted Norm. Rot. Factors	$\lambda_1 F_1^{(R\%)} - \lambda_2 F_2^{(R\%)} - \lambda_3 F_3^{(R\%)}$	-1	1	-1	1

Quadratic regression and ANN were identical in results obtained. All of the results with exception of “General”, summing the factor scores, and summing the normalized factor scores suggested robust settings of [-1 , 1]. To determine how “good”

these solutions were, the true LT values for each response was examined given each particular combination of settings, given in Table 23. The highlighted cell in each response column represents the optimal LT for that particular response, as shown in Table 18.

Table 23. LT Values for Factor Analysis Problem

x1	x2	Y1	Y2	Y3	Y4	Y5
-1	-1	78.61	73.99	481.43	-855.48	-695.80
-1	-0.5	42.18	37.67	468.17	-759.50	-559.49
-1	0	21.27	17.73	421.02	-692.68	-459.23
-1	0.5	10.50	8.07	345.55	-651.79	-388.05
-1	1	5.90	4.33	251.48	-634.68	-340.73
-0.5	-1	56.09	53.44	424.91	-692.08	-595.19
-0.5	-0.5	31.95	28.81	421.37	-606.28	-492.14
-0.5	0	18.59	15.75	385.28	-546.99	-418.94
-0.5	0.5	12.24	9.89	321.02	-510.89	-370.30
-0.5	1	10.51	8.61	237.18	-495.92	-342.40
0	-1	45.22	43.06	428.35	-572.06	-475.09
0	-0.5	29.51	26.59	433.90	-494.47	-403.77
0	0	21.64	18.61	406.12	-441.26	-356.60
0	0.5	19.42	16.49	348.23	-409.18	-329.74
0	1	22.06	19.36	267.61	-396.00	-321.11
0.5	-1	43.32	40.60	492.01	-487.45	-345.12
0.5	-0.5	34.19	30.48	507.99	-416.28	-302.19
0.5	0	31.75	27.52	487.78	-367.77	-278.11
0.5	0.5	35.38	30.82	433.38	-338.68	-270.60
0.5	1	45.92	41.27	350.98	-326.88	-279.02
1	-1	49.96	45.52	628.01	-432.27	-217.37
1	-0.5	47.58	41.71	657.77	-365.55	-197.54
1	0	52.51	45.44	646.34	-320.38	-191.76
1	0.5	65.75	57.60	594.52	-293.43	-199.21
1	1	89.71	80.79	507.33	-282.65	-220.71

Table 23 contains LT values based on different ranges depending on the response. To account for this, a percentage from optimal value was taken on each response. This was calculated by taking each LT value (in a particular response), and determined the percentage distance this value was from the optimal value (highlighted cell). Table 24 displays the percentage from optimal values. A column is appended on Table 24 to represent the average percentage distance a particular setting is from the optimal values.

Finally, this Table 24 is sorted on this column to represent the suggested true optimal settings for this problem.

Table 24. Percentage From Optimal LT Values

x1	x2	Y1	Y2	Y3	Y4	Y5	Average
-1	1	0.00	0.00	6.03	25.81	51.03	16.57
-0.5	1	78.07	98.95	0.00	42.03	50.79	53.97
-1	0.5	77.93	86.30	45.69	23.81	44.23	55.59
-0.5	0.5	107.42	128.39	35.35	40.28	46.78	71.64
-0.5	0	215.16	263.77	62.44	36.06	39.79	123.44
0	0.5	229.10	280.74	46.82	52.17	52.61	132.29
-1	0	260.48	309.40	77.51	19.03	34.00	140.08
0	1	273.89	347.08	12.83	53.71	53.85	148.27
0	0	266.80	329.74	71.23	48.42	48.75	152.99
0	-0.5	400.16	514.19	82.94	42.20	41.97	216.29
-0.5	-0.5	441.49	565.46	77.66	29.13	29.27	228.60
0.5	0	438.07	535.49	105.66	57.01	60.03	239.25
0.5	-0.5	479.46	604.04	114.18	51.34	56.57	261.12
0.5	0.5	499.73	611.79	82.72	60.41	61.11	263.15
-1	-0.5	614.86	769.93	97.39	11.22	19.59	302.60
0.5	-1	634.19	837.57	107.44	43.02	50.40	334.52
0.5	1	678.29	853.07	47.98	61.79	59.90	340.21
0	-1	666.49	894.48	80.60	33.13	31.72	341.28
1	-0.5	706.38	863.20	177.33	57.27	71.61	375.16
1	-1	746.74	951.30	164.78	49.47	68.76	396.21
1	0	790.05	949.46	172.51	62.55	72.44	409.40
-0.5	-1	850.74	1134.12	79.15	19.10	14.46	419.51
1	0.5	1014.47	1230.23	150.66	65.70	71.37	506.49
-1	-1	1232.44	1608.80	102.98	0.00	0.00	588.84
1	1	1420.53	1765.87	113.90	66.96	68.28	687.11

According to the results of the final column in Table 24, it was suggested that the settings [-1 , 1] were the true optimal settings for all five responses. Using these settings achieved the least difference from the optimal solutions for any response. According to Table 22, most of the derived factor analysis linear combinations determined these exact results.

The un-weighted factor scores, settings corresponding to [-1 , -1], were the only methods unable to obtain the optimal solution. This was most likely due to the fact that responses 4 and 5 should not have been held at the same weight as the other three. Finally, the “General” settings, [0.5 , 0.5], corresponded to the “middle of the road” suggested settings. For this example, using this method never achieved an optimal

solution for any response, but attempted to avoid performing consistently bad in any situation.

Although factor analysis was unable to reduce the five responses into a single dimension, the linear combination techniques proved useful. Therefore, it is suggested the use of factor analysis (and the linear combination techniques) as a mathematical approach to reducing high dimension response problems into a single dimension is superior to simply summing standardized response data (Davis, 2009).

IV. Application

4.1 Overview

In Chapter 3, research methodology was discussed. Examples of three different areas of research relating to robust parameter design were presented. The first area involved a search for new control variable settings that guard against system degradation. The second area utilized artificial neural networks rather than quadratic regression to fit highly nonlinear problems. Finally, factor analysis was implemented to reduce the dimensionality of multiple response problems.

These three techniques were applied to a computer algorithm developed by Johnson (2008). The computer algorithm, an autonomous global anomaly detector known as AutoGAD, has demonstrated usefulness in locating targets (anomalies) in hyper-spectral imagery (HSI). AutoGAD is currently employed using control settings suggested by Johnson, which were derived through experience. This research determined more robust (and doubly robust) settings than those currently implemented or previously researched, as in Davis (2009).

A background on hyper-spectral imagery, an explanation of AutoGAD, and results obtained from applying the new techniques to AutoGAD will be provided in this chapter.

4.2 Hyper-spectral Imagery

Hyper-spectral images are taken of an object or area of interest much like a digital photograph. The primary difference between the hyper-spectral images and digital

photographs lies within the region where the electromagnetic (EM) spectrum of the image is taken. The EM spectrum is displayed in Figure 24 for reference. The typical photograph produced from a common camera uses the visible part of the spectrum. This consists of a small number of bands; possibly just one band if dealing with black and white photographs. HSI utilizes the region from ultraviolet to infrared, as highlighted in Figure 25. The highlighted area consists of hundreds of bands, thus allowing for more information to be obtained about the object/area of interest.

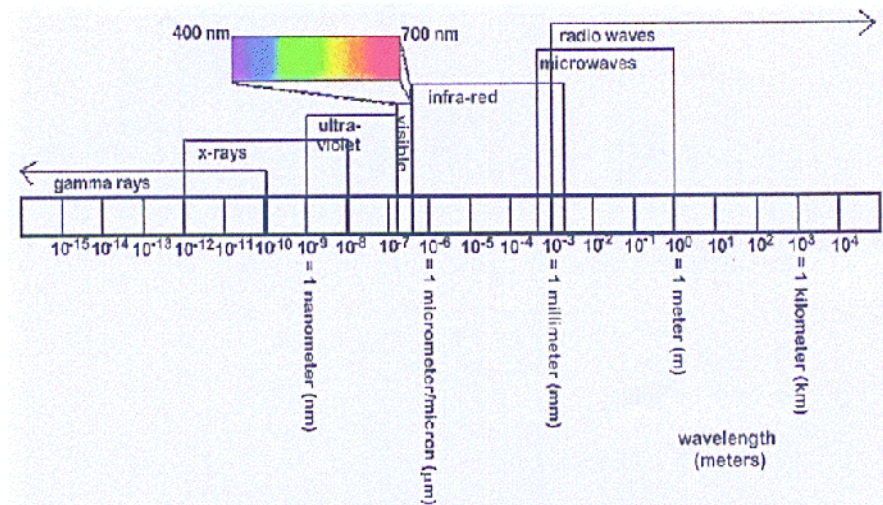


Figure 24. Electromagnetic Spectrum (Pabich, 2002)

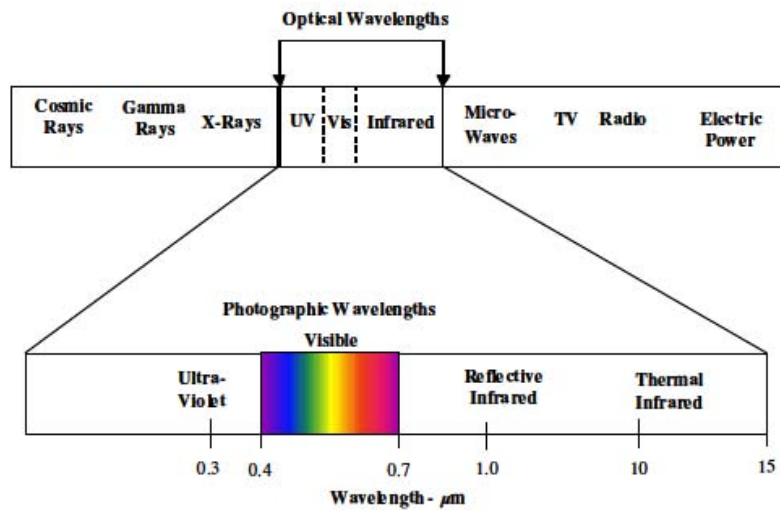


Figure 25. HSI range in EM spectrum (Landgrebe, 2003)

The capabilities of the HSI sensors are presented in detail by Pabich (2002). HSI sensors collect some form of reflected natural light, such as sunlight, from different objects. The energy of the reflected light is summarized into wavelength bins of the EM spectrum. Information given by the reflectance of objects allows for detection and identification.

Once the images are taken, a three dimensional HSI data cube is constructed, as seen in Figure 26. Viewing the cube from the spatial dimensions (two dimensional: i and j) is the same as viewing a photograph on a piece of paper. The spectral dimension (k) acts much like a stack of photographs on a table; each photograph is the same image but represents a different band of the EM spectrum. A representation of the spectral dimension (k) of a data cube is given in Figure 27.

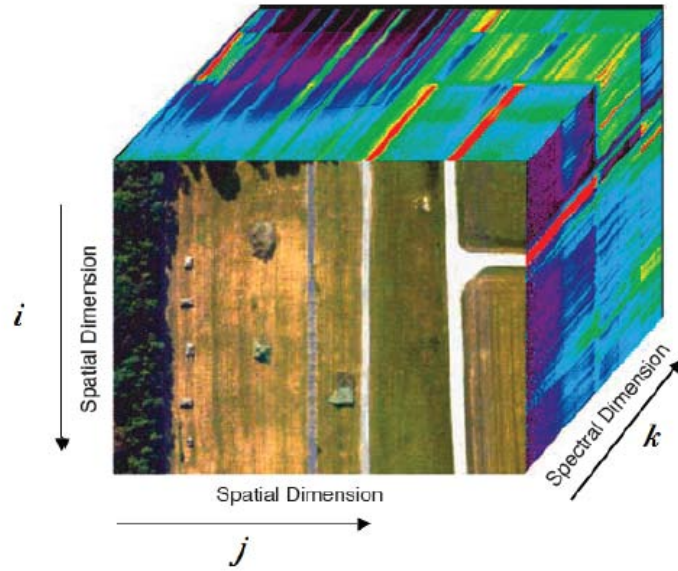


Figure 26. HSI Cube Example (Shaw et al., 2002)

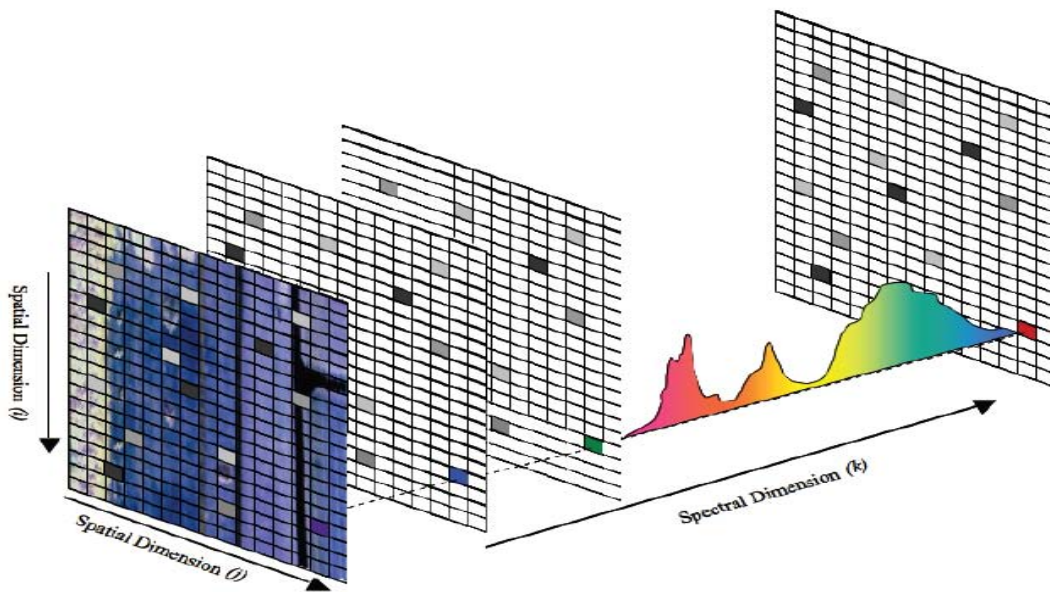


Figure 27. Layers of Data in Spectral Dimension (Miller, 2009)

To analyze the HSI data cube, the data in each image is converted into a two-dimensional matrix (Smetek, 2007). Suppose data from a hyper-spectral image that is

100 pixels wide by 75 pixels long contains 300 bands. This results in 7500 pixels with 300 spectral data points. Each pixel across the 300 spectral bands (7500 pixels) is converted into a single column in a new data matrix. The final data matrix size is 300 rows by 7500 columns, where each column represents a single pixel in the original image across all 300 spectral bands. This matrix is transposed to perform multivariate analysis. The process of transforming the HSI images into a matrix that can be utilized in numeric calculations (Miller, 2009) is depicted in Figure 28.

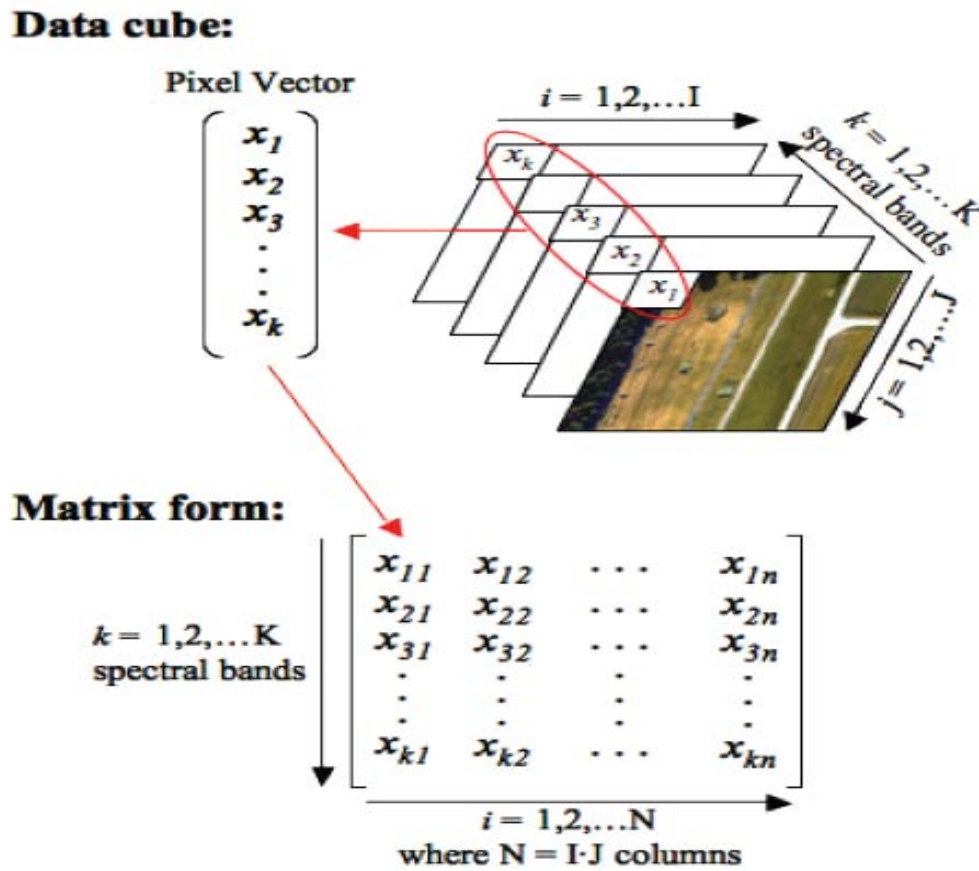


Figure 28. Transforming HSI Cube into Data (Miller, 2009)

4.3 AutoGAD

Johnson (2008) developed an autonomous global anomaly detector (AutoGAD) algorithm which provides information on the location of possible targets (anomalies) in real time. This computer algorithm utilizes the data matrix obtained from HSI data cubes to quickly and accurately locate possible targets within an image (Johnson, 2008; Davis, 2009; Miller, 2009). AutoGAD is written in MATLAB® and inputs HSI data to provide a resultant image of where targets are located. For testing purposes, AutoGAD has the capability to take “truth” images to determine the performance of the algorithm. A small discussion is provided on how AutoGAD works, and for a more complete understanding, the reader is referred to Johnson (2008).

After converting the HSI data into a two-dimensional matrix, as seen in Section 4.2, the algorithm was employed to assist the user in finding targets within the images, as depicted in Figure 29 (Johnson, 2008).



Figure 29. AutoGAD Algorithm

The first step applied principal components analysis (PCA) to the data in order to reduce the dimensionality. Once the dimensionality was reduced, the data was centered and scaled around zero with unit variance, which is known as whitening. Johnson (2008) proposed a Maximum Distance Secant Line (MDSL) to ascertain the amount of variance that should be retained in the dimensionality reduction following the whitening stage.

The second step sent the reduced data matrix through a process known as independent components analysis (ICA). ICA is not discussed here and the reader is referred to Johnson (2008) and its appropriate references.

Step three, feature selection, determined which objects were possible targets. A key assumption of AutoGAD is that the targets are rare in occurrence and are truly anomalies. Histograms were constructed to determine the frequency of the potential targets. Also, signal-to-noise ratio (SNR) statistics were employed to distinguish the targets from background noise. Thresholds were set to determine the pixels that fell out of the range of the background pixels. Miller (2009) extended this work which refined the thresholds for faster and better classification of potential targets.

The final step identified which pixels were indeed targets. If some target pixels were very close to what is referred to as the “zero bin,” an iterative adaptive noise (IAN) filtering technique was utilized by the algorithm for better distinguishing between the targets and background.

4.3.1 AutoGAD Outputs

The typical output for AutoGAD is a dark image with highlighted spots in different colors which represent the targets, or anomalies. Then, the user can distinguish where these targets were located by comparing the outputted images to the original images inputted into AutoGAD.

Alternatively, if the user possesses a “truth mask” to accompany the images, four outputs can be obtained from AutoGAD to determine the capability of the algorithm in detecting anomalies within images. The first output is “time,” which is measured in

seconds. This reflects the amount of time it takes for the algorithm to complete its “search” through the image. “True Positive Fraction” (TPF) is the second output which compares how well AutoGAD predicted detecting the anomalies against the “truth mask.” This value is obtained by taking the ratio of the number of pixels that AutoGAD correctly called targets (“T”) to the number of real target pixels (T) in the truth image or $P(T"/T)$. The truth value is in the denominator while the percentage of correct pixels identified by AutoGAD is in the numerator. TPF will always be a value between zero and one. The third output, “False Positive Fraction” (FPF), is computed by dividing the number of pixels incorrectly labeled targets (“T”) by the true number of non-target pixels (F) or $P(T"/F)$. Again, this value ranges between zero and one. Finally, “Target Fraction Percent” (TFP) is the ratio of true positives to the sum of true positives and false positives.

The objective of the algorithm is to accurately detect all anomalies in a quick manner (Johnson, 2008). Within AutoGAD, different settings have to be selected by the user which influences the four outputs of the algorithm. Specific combinations of settings cause increases/decreases in detection performance as well as processing time for different images. Different images also have an effect on the outputs as the optimal combination of settings for a particular image is not uniform for all images. Thus, robust parameter design (RPD) is desired to determine the best settings.

4.3.2 AutoGAD Control Variables

AutoGAD contains eleven controllable variables that need to be set prior to running the algorithm. Table 25 displays the control variable name, type, and range. For

detail on how each control variable works in AutoGAD, the reader is referred to Johnson (2008).

Table 25. Control Variables in AutoGAD

Control Variable	Type	Range
Dimension Adjust	Discrete	[-2 , 2]
Max Score Threshold	Continuous	[6 , 14]
Bin Width SNR	Continuous	[0.01 , 0.1]
PT SNR Threshold	Continuous	[1 , 6]
Bin Width Identify	Continuous	[0.01 , 0.1]
Smooth Iterations High	Discrete	[50 , 150]
Smooth Iterations Low	Discrete	[5 , 45]
Low SNR	Continuous	[4 , 14]
Window Size	Discrete	[3 , 11]
Threshold Both Sides	Discrete	[0 , 1]
Clean Signal	Discrete	[0 , 1]

4.3.3 AutoGAD Noise Variables

AutoGAD was a good candidate for RPD application not only because of the eleven control variables, but also because of the possibility of noise in the system. Davis (2009) first attempted to capture the noise within AutoGAD by stating the images themselves were noise. This was a reasonable suggestion since it is unknown what image will be sent through AutoGAD for detection purposes. Davis suggested that the noise variables were categorical. Although research exists for dealing with categorical noise variables (Brenneman & Myers, 2003), constructing the mean and variance models became complicated and involved prior probabilities. Here, an alternative method of modeling noise was considered in which continuous noise variables were implemented.

Three new noise variables were constructed for the AutoGAD algorithm by Mindrup *et al.* (2010). The noise variables required a “truth mask” to determine the appropriate output values for an image. The percentage of target pixels within each

image was considered as the first noise variable. This was calculated by taking the ratio of number of target pixels to the number of background pixels.

A Fisher's ratio was calculated for the second noise variable. Lohninger (1999) described the Fisher's ratio as a measure for the discriminating power of a particular variable. It attempted to portray the overlap of two distributions through mean (μ) and variance (σ^2). Class 1 was considered the target class and class 2 was the background class, as shown in Equation (4.1):

$$f = \frac{\mu_1^2 - \mu_2^2}{\sigma_1^2 + \sigma_2^2} \quad (4.1)$$

The final noise variable considered was the number of clusters in a given image. AutoGAD employed an X-means clustering algorithm recommended by Williams (2007) which determined the number of clusters by partitioning observations into different clusters based on their mean values. These three noise variables were developed to provide characteristics of an image rather than examining the image as a categorical variable. All three noise variables developed were continuous in nature allowing for a traditional RPD.

4.3.4 AutoGAD Setup Summary

A RPD was performed on AutoGAD to determine settings for the eleven control variables based on the three suggested noise variables. Eight images, as well as their "truth masks," were provided to the author for research on AutoGAD. To obtain more data, each of the eight images was divided in half allowing for 16 images to be analyzed. When performing RPD, eight half-images were used for training (quadratic regression or

artificial neural networks) and the other eight for testing, verification, and validation. This method allowed for a holdout set which determined how results would look if the RPD settings were implemented on images never before seen, thus providing more real-life scenario results.

This method differed from Johnson (2008) and Davis (2009) who each developed control variable settings based on knowledge of all 16 half-images. This can color the results since no new information was presented in AutoGAD to determine its robustness to new images. This research combated this problem through utilizing the holdout method discussed above.

4.4 AutoGAD and RPD

As explained in Section 4.3.1, AutoGAD contained four different output values for detection: Time, TPF, FPF, and TFP. This constituted a four response problem in which traditional quadratic regression and ANNs were applied to the AutoGAD data sets to derive mean and variance models for each output. These results were then compared to demonstrate the benefit of using ANNs when quadratic regression fails to model the problem appropriately.

As will be shown, all four responses did not share the same combination of optimal robust settings. This warranted the use of RPD with multiple responses to combine the four responses into a single dimension. A discussion is provided in Section 4.5 on the techniques used to formulate a model(s) which considered all four responses simultaneously in AutoGAD.

4.4.1 AutoGAD RPD Design

To conduct a RPD, an appropriate design was constructed to intelligently collect data from AutoGAD. A full factorial design with 11 control variables and three noise variables was too large of a design to choose due to the time required to collect data. As shown in the example problems of Chapter 3, combined array designs were appropriate when constructed correctly. Two methods of performing ANNs that involved “pre-processing” and “post-processing” were presented in Section 3.4.1. The former required a crossed array design to determine appropriate mean and variance values across noise settings. Therefore, a central composite design (CCD) on the control variables was crossed with a 2^3 factorial design on the noise variables. The resultant design contained 2160 CCD runs crossed with eight noise runs resulting in 17280 design points.

In the CCD, nine of the control variables were varied over five levels: One center point, one at the plus and minus factorial points, and one at the plus and minus face-centered points (same as the factorial). These nine control variables were tested at each plus and minus factorial point of the remaining two control variables, since only a range of [0,1] was utilized on these two controls. This CCD allowed an appropriate number of samples taken from the large design space of the control variables. A sample segment of the CCD is given in Appendix B. This design was large, but since the AutoGAD algorithm operated relatively quickly, collecting this amount of data was not too costly in terms of time or money. Finally, this design allowed for adequate comparison of quadratic regression versus the ANNs.

For this research, all settings were in terms of coded values. This resulted in every control or noise variable setting to range from [-1 , 1]. To code natural (uncoded)

values, Equation (4.2) was used (Myers & Montgomery, 2002), where ξ represented the i th natural setting of the j th variable to obtain the coded (x) setting. Below, $\max(\xi_{.j})$ represented the maximum value for the j th variable across all rows (i) of natural settings:

$$x_{ij} = \frac{\xi_{ij} - \left(\frac{\max(\xi_{.j}) + \min(\xi_{.j})}{2} \right)}{\left(\frac{\max(\xi_{.j}) - \min(\xi_{.j})}{2} \right)} \quad (4.2)$$

Following the analysis of RPD, the optimal settings were calculated in coded terms. These values were then converted back to their natural settings by applying Equation (4.3):

$$\xi_{ij} = \left[x_{ij} * \left(\frac{\max(\xi_{.j}) - \min(\xi_{.j})}{2} \right) \right] + \left(\frac{\max(\xi_{.j}) + \min(\xi_{.j})}{2} \right) \quad (4.3)$$

4.4.2 AutoGAD Quadratic Regression RPD

Once the CCD was set up, data was collected in AutoGAD based on the design points for all four response values for each run. At this point, AutoGAD was separated into four different problems, one for each response. Box-Cox (Myers & Montgomery, 2002) analysis was applied to the resultant response values which determined the need for any transformations. A lambda value, which represents the power to which the response data is raised based on the Box-Cox transformation, was obtained and response values were re-calculated. This transformed data raised the R^2 statistic and provided a better fit.

Quadratic regression was applied to each of the four outputs separately and an overall process model for each response was calculated. The overall process model was derived by calculating $(X'X)^{-1}(X'Y)$ where X represented the design with a column of ones added to the beginning and Y represented the response examined. This process model became quite large containing one intercept term, 11 control main effects, 11 control quadratic effects, 55 control by control interaction effects, three noise main effects, and 33 control by noise interaction effects, totaling 114 terms. As previously discussed, no noise by noise interactions were considered, based on suggestions by Myers & Montgomery (2002). However, including the noise by noise interaction would have increased the R^2 values and provided better fits.

Many of the 114 terms were insignificant according to the Analysis of Variance (ANOVA) table; therefore, a backward stepwise regression approach was employed to reduce the model. Once the full model was obtained, an ANOVA analysis was performed and the term with the highest p-value (assuming a p-value > 0.10) was removed from the model. The process model was then recalculated to obtain new coefficients and the term with the highest p-value was removed. This process was continued until a reduced model was developed that contained only significant terms (p-value ≤ 0.10). As a side note, the control main effects and the noise main effects were never removed to maintain hierarchy and to establish mean and variance models for analysis. If a main effect was removed, this meant that the settings for that particular control variable would have no effect on the overall response or variance of the response. Due to the sizes of the reduced models, the models are not presented in this document.

Once the reduced model was obtained, it was then separated into its appropriate mean and variance models as shown in Equations (2.7) and (2.8). These two equations were employed as dual responses in an effort to minimize the variance and satisfy constraints on the mean. Also, it is important to note that this research assumed continuous control and noise variables. This dual response problem was solved using the LT formulations in Equation (4.4):

$$\begin{aligned}
MSE_{\min} &= \{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)] \\
MSE_{\max} &= -\{\hat{u}_z[\hat{y}(x, z)]\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)] \\
MSE_{target} &= \{\hat{u}_z[\hat{y}(x, z)] - T\}^2 + \hat{\sigma}_z^2[\hat{y}(x, z)]
\end{aligned} \tag{4.4}$$

Time and FPF responses were minimized while TPF and TFP were maximized. All instances in Equation (4.4) required the function to be minimized. To search for the optimal LT value and its settings, a complete enumeration of integer control variables and a coarse discretization of the remaining control variables was performed. This resulted in 320,000,000 combinations of the 11 control variables. MATLAB® calculated LT values of this enumeration set in less than two hours. Other, possibly quicker, methods could be employed; however, this enumeration technique was utilized to avoid falling into local minimum solutions.

After obtaining the LT values for all possible combinations, the control settings associated with the minimum LT value were chosen to represent the robust parameters. The optimal control settings for the four different outputs tested are reported in Table 26, as well as the expected mean, variance, and LT values. These control settings should prove robust to new images introduced into AutoGAD.

Table 26. Optimal Settings for 4 Responses Suggested by QR

Control	Time	TPF	FPF	TFP
DA	2	-2	-2	-2
MST	6	6	6	6
BWSNR	0.01	0.01	0.01	0.08
PTSNR	1	1	1	5
BWI	0.1	0.01	0.01	0.1
TBS	0	0	0	0
CS	1	0	0	1
SIH	150	50	50	150
SIL	45	45	5	5
LSNR	14	14	4	14
WS	3	3	5	11
Outputs				
Pred. Mean	3.05	1.102	0.13	1.12
Pred. Var	12.033	0.189	0.45	0.398
Pred. LT	21.34	-1.026	0.4669	-0.8659

As seen in Table 26, none of the four responses shared the same optimal robust settings. More importantly, the four responses only agreed on the same setting for one of the eleven control variables: MST. The mean results for TPF and TFP indicated values higher than one, which was infeasible. This indicated the lack of fit qualities these quadratic models possessed. To test the validity of the results in Table 26, these settings were applied to the eight untested images. The results for each respective response were averaged across the eight images to establish a mean and variance. Finally, to maintain consistency with the selection of the settings, the LT values were calculated. The results when the suggested optimal settings utilizing quadratic regression were tested are presented in Table 27.

Table 27. Testing Results for Optimal Settings in QR

	Mean	Variance	LT
Time	39.0214	1590.1000	3112.80
TPF	0.9752	0.0004	-0.9507
FPF	0.1180	0.0030	0.0170
TFP	0.8739	0.0431	-0.7207

Prior to explanation of the results in Table 27, it is important to note that the LT values for TPF and TFP were negative. This result was expected due to the construction of a maximized response LT problem in Equation (4.4). The designation of a negative sign on the mean-squared term of the LT drove the LT values into the negative response realm. The LT value was still minimized; therefore, the settings with the largest negative value were optimal.

The results given in Table 27 appear to be appropriate. TPF and TFP obtained rather large mean values while maintaining low variance. The mean of FPF was higher than desired, but contained a very small variance value. Finally, time appeared to do well in terms of mean but calculated an extremely large variance value.

To aid in understanding the reasons time and FPF did not achieve great results, all four ANOVA tables and residual plots were examined. Table 28 - Table 31 display the ANOVA tables for each response. As seen in the tables, each model was significant; however, lack of fit was prevalent. The strong significance in the lack of fit indicated the quadratic regression model did not accurately fit the data, which led to false conclusions.

A coefficient of determination, or R^2 , and its derivatives were utilized to assist in the explanation of ANOVA. R^2 , the amount of variability accounted for in the data, was calculated as (Montgomery *et al.*, 2004):

$$R^2 = 1 - \frac{SS_E}{SS_T} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2 / n} \quad (4.5)$$

where SS_E was the sum of squares of residuals, SS_T was the total sum of squares, y_i represented the true response values, and \hat{y}_i the fitted response values for n treatment rows (sample size).

The R^2 statistics increased as more terms were added to the model. Another statistic, the adjusted R^2 , only increased if the added terms to the model reduced the MSE value. This statistic, for p terms (regressors), was calculated as:

$$adj(R^2) = 1 - \frac{SS_E / (n - p)}{SS_T / (n - 1)} \quad (4.6)$$

Finally, a predicted R^2 statistic was utilized which gave an indication on how well the model predicted new response data. This is similar to R^2 , except for each residual (i), the model was fit to the remaining $n-1$ sample, thus taking out the i th residual. This type of residual analysis is known as the PRESS residual and is denoted as $\hat{y}_{(i)}$. This predicted R^2 was computed as:

$$pred(R^2) = 1 - \frac{PRESS}{SS_T} = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_{(i)})^2}{\sum_{i=1}^n y_i^2 - \left(\sum_{i=1}^n y_i \right)^2 / n} \quad (4.7)$$

These three R^2 statistics were examined in the ANOVA tables of each response. In terms of time (Table 28), an R^2 value of 0.37 was obtained. This value was extremely low, which may help to explain the poor results reported in Table 27. The

other three responses obtained R^2 values around 0.61 or 0.69. These values indicated some ability quadratic regression had in fitting the given data. Their predicted R^2 values fell in the same general range, thus allowing for decent prediction of optimal control settings given new data. However, great results were not expected due to the lack of fit significance.

Table 28. ANOVA for Time

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	340.70	36	9.46	285.5	< 0.0001	significant
Residual	571.70	17246	0.03			
Lack of Fit	570.01	16950	0.03	5.873	< 0.0001	significant
Pure Error	1.69	296	0.01			
Cor Total	912.41	17282				
R-Squared	0.3734					
Adj R-Squared	0.3721					
Pred R-Squared	0.3708					

Table 29. ANOVA for TPF

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	1435.25	60	23.92	451.40	< 0.0001	significant
Residual	912.90	17227	0.05			
Lack of Fit	911.84	16931	0.05	14.95	< 0.0001	significant
Pure Error	1.07	296	0.00			
Cor Total	2348.15	17287				
R-Squared	0.6112					
Adj R-Squared	0.6099					
Pred R-Squared	0.6086					

Table 30. ANOVA for FPF

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	9105.15	72	126.46	537.95	< 0.0001	significant
Residual	4046.90	17215	0.24			
Lack of Fit	4046.36	16919	0.24	130.79	< 0.0001	significant
Pure Error	0.54	296	0.00			
Cor Total	13152.05	17287				
R-Squared	0.6923					
Adj R-Squared	0.6910					
Pred R-Squared	0.6899					

Table 31. ANOVA for TFP

Source	Sum of Squares	df	Mean Square	F Value	p-value Prob > F	
Model	1840.41	62	29.68	438.30	< 0.0001	significant
Residual	1166.56	17225	0.07			
Lack of Fit	1166.07	16929	0.07	41.17	< 0.0001	significant
Pure Error	0.50	296	0.00			
Cor Total	3006.97	17287				
R-Squared	0.6120					
Adj R-Squared	0.6107					
Pred R-Squared	0.6095					

Figure 30 displays the normal probability plots for each response: Time (top-left), TPF (top-right), FPF (bottom-left), and TFP (bottom-right). As seen in the first plot (time), there was a heavy-tailed distribution. This phenomenon caused the low R^2 value and significant lack of fit. Quadratic regression was not suggested for this response. The normal probability plots for the other three responses all indicated a light-tailed distribution, which indicated a slight diversion from the normality assumption crucial to ANOVA. However, the residuals fell near the line and decent R^2 values were maintained.

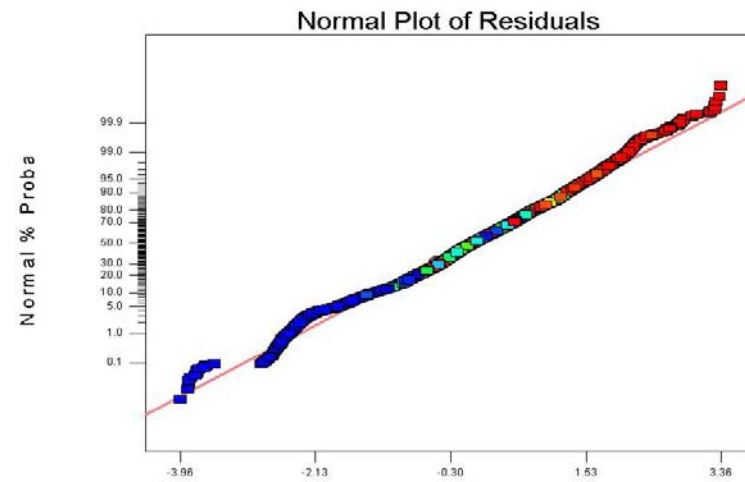
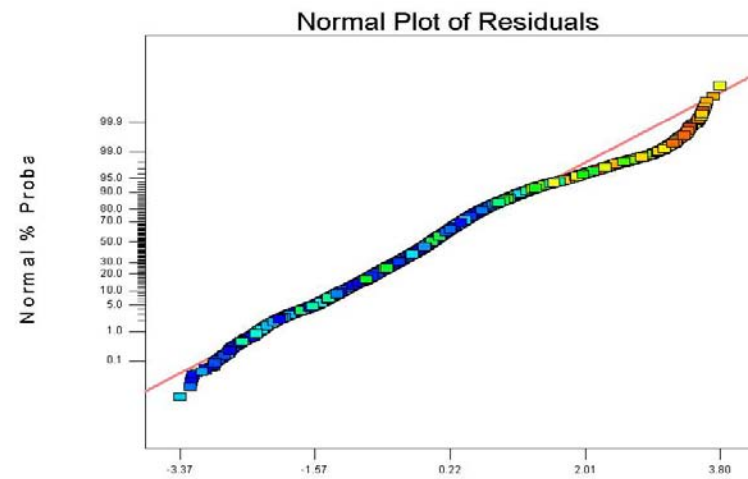
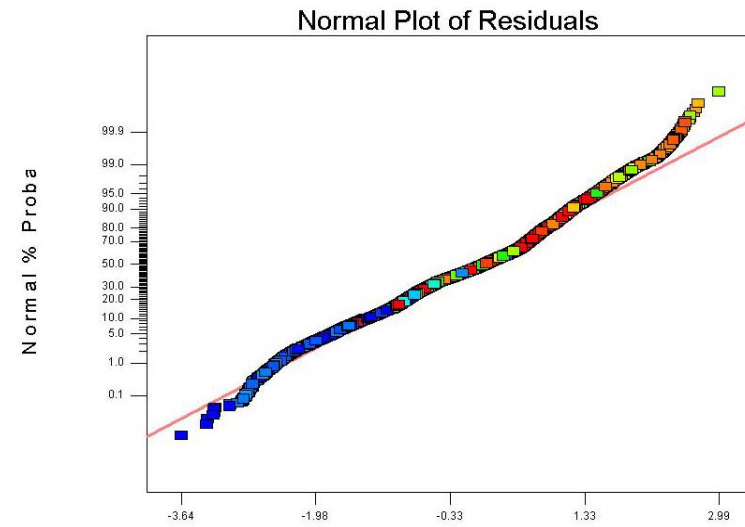
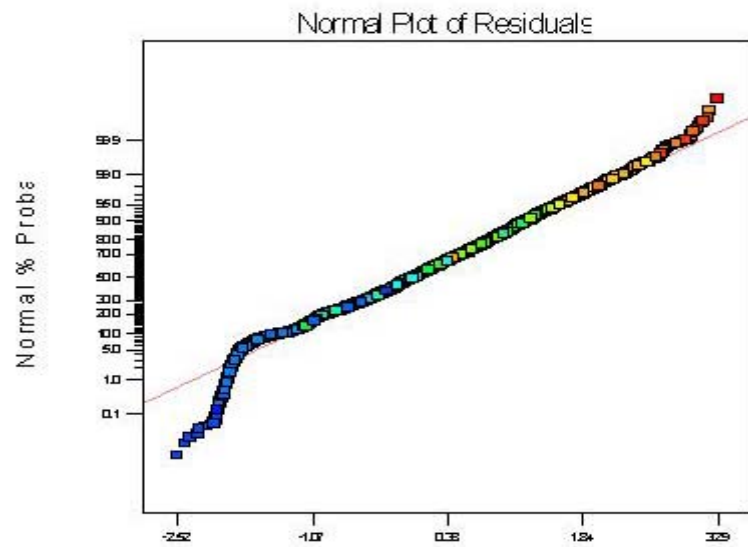


Figure 30. Residual Plots for Time, TPF, FPF, TFP

4.4.3 AutoGAD ANN RPD

Traditional RPD methodology was demonstrated on single response problems using quadratic regression, as shown in Section 4.4.2. As discussed in the previous section, the R^2 values were low and the lack of fit was significant for all four responses. Therefore, quadratic regression failed to properly fit the data, which produced possible non-optimal robust settings as a result. Artificial neural networks (ANNs) were instituted at this juncture to provide better fits to these seemingly nonlinear problems.

In order to construct the ANN to fit the data, the ANN needed to be trained properly. To train the neural network, a hold-out method was utilized. This method withheld one-third of the data for testing of the neural network and employed two-thirds for training (Kuncheva, 2004). Withholding data from training allowed for the spread parameter of the network to be adjusted appropriately to its optimal setting. The withheld data was tested at each spread increment and the root mean square error (RMSE) was calculated on the test set. The spread value with the lowest RMSE was chosen and the network was considered trained. An example spread versus RMSE plot is given in Figure 31. Since two outputs (mean and variance) were calculated, a spread value and RMSE was attributed to each output. Typically, the optimal spread was the same for each output; however, if they differed, a tradeoff assessment was necessary. The plot in Figure 31 displays the spread versus RMSE for mean and variance for the output TPF. The optimal spread value was 0.85 for this response.

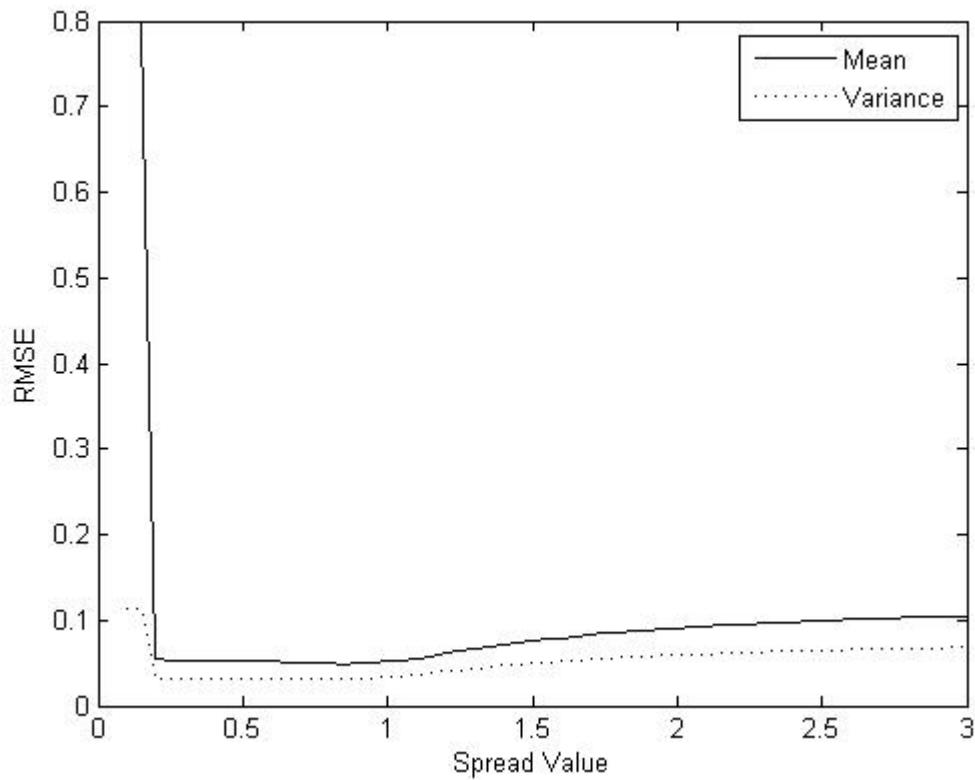


Figure 31. Spread vs. RMSE for TPF

The hold-out method employed caused the ANNs to be at a disadvantage with respect to quadratic regression. When training the network, ANNs only utilized two-thirds of the data; whereas, quadratic regression utilized the entire data set for training. Although at a disadvantage, better results were obtained using ANNs.

Since a crossed array design was implemented on AutoGAD, the “pre-processing” ANN approach was employed due to its ability to fit a network in less time than the “post-processing” approach in the absence of noise variables as inputs. This reduced the number of inputs from 14 to 11, which decreased the network calculation time. As depicted previously in Figure 16, only the control variables were inputted and mean and variance values were outputted for every control variable combination. For

each mean and variance value with respect to each unique combination of control variable settings, an LT statistic was calculated.

In Section 4.4.2, a summary of the enumerated search performed on the setting space to determine optimal LT solutions was provided. The same principle was applied at this point, but due to the construction of the MATLAB® code for neural networks, a coarser and reduced discretized data set was utilized. The testing set was reduced to 4,000,000 possible combinations as a result; however, an appropriate number of settings was tested for each control variable.

Implementing the smaller exhaustive search placed the ANN at another disadvantage compared to quadratic regression. First, the neural network used less data (only two-thirds) to train and fit appropriate mean and variance models. Now, fewer possible combinations were searched in the LT space due to time restrictions. However, the example problem tested in Section 3.4.2 employed the same hold-out principle and reduced enumerated search, yet yielded superior results to quadratic regression.

The four response problems were examined separately and optimal robust settings were obtained for each response. The results obtained using RBFNNs are reported in Table 32 and the optimal robust settings obtained using GRNNs are reported in Table 33. Along with the settings, the predicted mean, variance, and LT values are displayed from training the ANNs.

Table 32. Optimal Settings for Four Responses (RBFNN)

Control	Time	TPF	FPF	TFP
DA	-2	0	-2	-2
MST	14	6	6	6
BWSNR	0.1	0.07	0.1	0.1
PTSNR	6	3.5	6	6
BWI	0.01	0.04	0.1	0.1
TBS	0	1	0	0
CS	1	1	1	1
SIH	50	83	150	150
SIL	45	25	45	45
LSNR	4	11	14	14
WS	3	3	11	11
Outputs				
Pred. Mean	6.9526	0.9933	0.0003	0.9058
Pred. Var	82.438	0.0002	6.278E-07	0.0529
Pred. LT	130.77	-0.9865	7.238E-07	-0.7676

Table 33. Optimal Settings for Four Responses (GRNN)

Control	Time	TPF	FPF	TFP
DA	-2	2	-2	-2
MST	14	6	14	6
BWSNR	0.1	0.01	0.1	0.1
PTSNR	1	6	1	6
BWI	0.01	0.01	0.1	0.1
TBS	1	1	0	1
CS	0	0	1	1
SIH	150	50	50	150
SIL	45	5	45	45
LSNR	4	4	14	14
WS	11	11	11	11
Outputs				
Pred. Mean	6.61	0.9864	0.0007	0.9154
Pred. Var	83.625	0.001	3.68E-06	0.0479
Pred. LT	127.31	-0.972	4.212E-06	-0.7901

To determine the validity of using ANNs over quadratic regression, the settings in Table 32 and Table 33 were tested in AutoGAD on the 8 un-trained images. Mean and variance were calculated for each combination of settings across all images tested. From

the mean and variance, the LT statistic was calculated. The LT statistic was employed to maintain consistency since it was used to determine the optimal robust settings. The results for RBFNNs are reported in Table 34 and GRNN results are displayed in Table 35 for the four responses.

Table 34. Testing Results for Optimal Settings using RBFNN

	Mean	Variance	LT
Time	29.4	742.9	1606.5
TPF	0.9644	0.0019	-0.9282
FPF	0.0005	0.0000	0.0000
TFP	0.9158	0.0226	-0.8161

Table 35. Testing Results for Optimal Settings using GRNN

	Mean	Variance	LT
Time	24.1	662.7	1245.7
TPF	0.9806	0.0010	-0.9606
FPF	0.0002	0.0000	0.0000
TFP	0.9074	0.0232	-0.8002

As seen in Table 34 and Table 35, appropriate results were obtained. To measure the “goodness” of these results, they were compared to those obtained using quadratic regression. A comparison of the settings for quadratic regression and the ANNs is shown in Table 36. The results obtained when testing the robust settings when quadratic regression and ANNs were used are displayed in Table 37 - Table 40.

Table 36. Optimal Settings for QR and ANNs

	Time			TPF			FPF			TFP		
	QR	RBFNN	GRNN	QR	RBFNN	GRNN	QR	RBFNN	GRNN	QR	RBFNN	GRNN
Control												
DA	2	-2	-2	-2	0	2	2	-2	-2	-2	-2	-2
MST	6	14	14	6	6	6	6	6	14	6	6	6
BWSNR	0.01	0.1	0.1	0.01	0.07	0.01	0.01	0.1	0.1	0.08	0.1	0.1
PTSNR	1	6	1	1	3.5	6	1	6	1	5	6	6
BWI	0.1	0.01	0.01	0.01	0.04	0.01	0.01	0.1	0.1	0.1	0.1	0.1
TBS	0	0	1	0	1	1	1	0	0	0	0	1
CS	1	1	0	0	1	0	0	1	1	1	1	1
SIH	150	50	150	50	83	50	50	150	50	150	150	150
SIL	45	45	45	45	25	5	45	45	45	5	45	45
LSNR	14	4	4	14	11	4	14	14	14	14	14	14
WS	3	3	11	3	3	11	3	11	11	11	11	11

Table 37. Testing Results for QR and ANNs on Time

	Time		
	Mean	Variance	LT
QR	39.02	1590.1	3112.8
RBFNN	29.39	742.9	1606.5
GRNN	24.15	662.7	1245.7

Table 38. Testing Results for QR and ANNs on TPF

	TPF		
	Mean	Variance	LT
QR	0.9752	0.0004	-0.9507
RBFNN	0.9644	0.0019	-0.9282
GRNN	0.9806	0.0010	-0.9606

Table 39. Testing Results for QR and ANNs on FPF

	FPF		
	Mean	Variance	LT
QR	0.1991	0.0070	0.0466
RBFNN	0.0005	0.0000	0.0000
GRNN	0.0002	0.0000	0.0000

Table 40. Testing Results for QR and ANNs on TFP

	TFP		
	Mean	Variance	LT
QR	0.8739	0.0431	-0.7207
RBFNN	0.9158	0.0226	-0.8161
GRNN	0.9074	0.0232	-0.8002

As seen in Table 36, extremely different settings were obtained using quadratic regression and ANNs, with the possible exception of TFP. This indicated that the two methods of modeling the mean and variance were quite different.

In terms of time (Table 37), ANNs outperformed quadratic regression. The ANNs reduced the average time by nearly 15 seconds and cut the variance value in half. GRNNs obtained a better result than the RBFNNs by obtaining a 22.5 percent reduction in LT value with respect to the LT obtained in quadratic regression. These results

suggested that ANNs were clearly preferred to quadratic regression, especially since an R^2 value of 0.37 was obtained.

TPF displayed the only possible scenario where quadratic regression was competitive. As seen in Table 38, quadratic regression obtained a lower (higher negative) LT value than RBFNNs, but GRNNs were still preferred by obtaining the lowest LT value. Although RBFNNs performed the worst in this scenario, their results remained competitive since they achieved similar means with slightly larger variances.

FPF conveyed strong results for ANNs as shown in Table 39. An LT value of almost zero was obtained when either ANN was used, which provided a nearly perfect LT value. Quadratic regression maintained low variance, but the mean value ballooned to 0.11, which indicated the presence of multiple false positive identifications.

Finally, the results shown in Table 40 represent superior results, in general, for ANNs as compared to quadratic regression for TFP. Compared to quadratic regression, the RBFNNs and GRNNs saw an increased mean of 4.8 and 3.8 percent, respectively, while the variance was reduced. This resulted in lower LT values for each ANN.

The results presented in Table 37 – Table 40 provided strong evidence in the usefulness of ANNs when quadratic regression fails to properly fit the data. The extreme case was seen with time, since an R^2 value of 0.37 was obtained. However, the other three responses achieved an R^2 value in the range of 0.60-0.70, yet ANNs still performed better or the same as quadratic regression. When dealing with problems containing highly nonlinear responses, as suggested in the AutoGAD problem, ANNs should be considered as an alternative to traditional quadratic regression.

4.5 AutoGAD and System Degradation

Optimal robust settings for the four responses of AutoGAD using quadratic regression and ANNs were presented in Sections 4.4.2 and 4.4.3. These settings were expected to be robust under normal operating conditions of the algorithm. However, new robust settings were necessary to guard against system degradation. In this section, the performances of doubly robust settings are examined.

As explained in Section 3.3, system degradation in software can be represented by being exposed to inputs beyond its experience and training. In AutoGAD, this equated to a new image considered to be noisier than any image on which training was performed. A signal-to-noise ratio (SNR) was calculated for all eight available images. The image with the largest SNR value was selected as the “noisiest” image. This corresponded to Image 6 of the available images. If this image was withheld from AutoGAD training, it was expected that AutoGAD would perform relatively poorly with respect to this image. That is to say, degraded performance was expected. It is important to note that although this image was the “noisiest” it was not necessarily much noisier than the other images. In other words, it was not truly an outlier compared to the other seven images. As a side note, all analysis performed in the previous sections contained both halves of Image 6 in the testing set; therefore, the same training data was utilized in this section.

For simplicity, only two responses were utilized to test AutoGAD under system degradation using both quadratic regression and ANNs. TPF and TFP were chosen as the responses, since both quadratic regression and ANNs performed well in determining their appropriate optimal settings. Time was not considered due to the substantial lack of fit

that quadratic regression displayed for this response. Also, FPF was not chosen since values of zero were obtained for nearly every image.

4.5.1 Doubly Robust Settings in Quadratic Regression

Quadratic regression was presented in Section 4.4.2 in which optimal robust settings for TPF and TFP, as given in Table 26, had associated LT values of $Y^* = -0.9507$ and $Y^* = -0.7207$, respectively. Recall, these values were negative due to both responses being maximized. To determine the doubly robust settings, the algorithm outlined in Figure 10 was employed.

The doubly robust algorithm was conducted until $Y^{**} = -0.7605$ and $Y^{**} = -0.5766$ was realized, which was a 20 percent increase in LT. x^* was equal to the settings given in Table 26 corresponding to TPF and TFP. The coefficients (C^{old}) to begin the gradient search corresponded to the coefficients used to construct the LT statistic to solve for x^* . Once the coefficients were obtained, the derivative of LT (Y) was taken with respect to these coefficients (C), $\nabla Y = \frac{\partial Y}{\partial C} \Big|_{C^{old}}$. A step was then taken in the gradient direction and the new dual response problem was solved to find the minimum LT statistic and its corresponding control variable settings. This process was repeated until Y^{**} was realized. The optimal control variable settings associated with Y^{**} were considered to be doubly robust settings. These settings should prove robust to noise variables as well as robust to system degradation.

The doubly robust settings and original optimal robust settings were tested against Image 6 which represented system degradation in AutoGAD. Eight replications of the

two combinations of settings were tested against both halves of Image 6 to obtain a mean, variance, and LT value. The original robust settings as well as the doubly robust settings for output TPF are displayed in Table 41. The results of these settings in TPF against Image 6 are reported in Table 42.

Table 41. Original and Doubly Robust Settings for TPF

Control	TPF	
	Original	Doubly Robust
DA	-2	-2
MST	6	7
BWSNR	0.01	0.09
PTSNR	1	3
BWI	0.01	0.01
TBS	0	1
CS	0	0
SIH	50	150
SIL	45	45
LSNR	14	8
WS	3	3

Table 42. Image 6 Results for TPF Settings

TPF - Quadratic Regression			
	Mean	Variance	LT
Original	0.9627	0.0005	-0.9263
Doubly Robust	0.9634	0.0004	-0.9278

The results shown in Table 42 validated the use of doubly robust settings in the presence of system degradation. By moving to a 20 percent increase in LT, the doubly robust settings guarded against images that caused degradation. The doubly robust settings obtained a larger mean (desired) and smaller variance than the original robust settings for Image 6. Recall, Image 6 was not necessarily a very noisy image, which indicates why the results did not differ as much as possibly expected. This however changed when ANNs were examined.

The same analysis was performed on TFP, with a slight change in the gradient step. Originally, a 20 percent increase in LT was utilized, but this step size was deemed too large of an increase for this problem. A substantial increase (not desired) in LT value was obtained with the doubly robust settings. This step size will vary from problem to problem. Therefore, instead of a 20 percent increase in LT, a five percent increase in LT was examined. The settings for both the original and doubly robust with respect to TFP are displayed in Table 43 and the results obtained when these settings were applied to Image 6 are reported in Table 44. Recall that eight replications were performed on Image 6 to calculate a mean, variance, and LT statistic for each combination of settings.

Table 43. Original and Doubly Robust Settings for TFP

Control	TFP	
	Original	Doubly Robust
DA	-2	-2
MST	14	14
BWSNR	0.1	0.1
PTSNR	6	6
BWI	0.08	0.09
TBS	0	0
CS	1	1
SIH	150	150
SIL	5	5
LSNR	14	14
WS	3	11

Table 44. Image 6 Results for TFP Settings

TFP - Quadratic Regression			
	Mean	Variance	LT
Original	0.9206	0.0110	-0.8365
Doubly Robust (5%)	0.9934	0.0004	-0.9864

Once again, the results portrayed that doubly robust settings were superior to original robust settings when faced with system degradation. In fact, applying doubly

robust settings to Image 6 in terms of TFP yielded almost perfect results and increased the LT value significantly. These results demonstrated the utility of doubly robust settings over typical robust settings to guard against system degradation. The same analysis using ANNs rather than quadratic regression was applied to system degradation, as explained in the next section.

4.5.2 Doubly Robust Settings in ANNs

ANNs were shown to be the preferred choice over quadratic regression in Section 4.4.3. In what follows, doubly robust settings were also calculated using the gradient method outlined in Figure 23. The same responses, TPF and TFP, were examined to determine doubly robust settings for RBFNNs and GRNNs. Also, the same 20 percent increase in LT for TPF and five percent LT increase for TFP were maintained.

Gradient analysis was applied to RBFNNs on the response TPF. The original robust settings and the doubly robust settings are displayed in Table 45 for comparison. The results obtained when these two settings were applied to both halves of Image 6 over eight replications are reported in Table 46. As seen in this table, the doubly robust settings achieved a better LT value. Approximately a one percent increase was seen in the mean while a slight reduction of variance was observed, which led to the lower LT statistic collected with the doubly robust settings. This further warranted the use of doubly robust settings in the presence of possible system degradation.

Table 45. RBFNN Original and Doubly Robust Settings for TPF

Control	TPF	
	Original	Doubly Robust
DA	0	0
MST	6	6
BWSNR	0.07	0.07
PTSNR	3.5	3.5
BWI	0.07	0.04
TBS	0	0
CS	1	1
SIH	83	83
SIL	25	25
LSNR	11	11
WS	3	3

Table 46. RBFNN Image 6 Results on TPF

TPF - RBFNN			
	Mean	Variance	LT
Original	0.9649	0.0005	-0.9304
Doubly Robust	0.9774	0.0004	-0.9549

Gradient analysis was then applied to GRNNs on TPF. The settings for the original robust settings and doubly robust settings types are displayed in Table 47 and the results when applied to Image 6 are reported in Table 48. This was one of the only instances in which the doubly robust settings failed to outperform the original settings. As depicted in Table 48, the original robust settings were nearly perfect in determining all targets within the image. This situation made it difficult for any possible new settings to outperform the original settings. Although the doubly robust settings failed to outperform the original robust settings, their difference was minimal. Less than a one percent difference was observed between the means of each solution and the LT statistic of each solution. Therefore, either combination of these settings were deemed appropriate for this new image.

Table 47. GRNN Original and Doubly Robust Settings on TPF

Control	TPF	
	Original	Doubly Robust
DA	2	-2
MST	6	6
BWSNR	0.1	0.1
PTSNR	6	1
BWI	0.01	0.01
TBS	0	1
CS	1	1
SIH	150	50
SIL	5	5
LSNR	14	14
WS	3	3

Table 48. GRNN Image 6 Results on TPF

	TPF - GRNN		
	Mean	Variance	LT
Original	0.9941	0.0000	-0.9881
Doubly Robust	0.9891	0.0000	-0.9782

As in quadratic regression, only TPF and TFP were examined. Time resulted in large variance values and a significant lack of fit with quadratic regression. FPF consistently maintained a mean value near zero for all images, making degradation difficult to capture. Therefore, to remain consistent with quadratic regression, the response TFP was the only other response examined. First, RBFNN gradient analysis was applied to this response. The original robust settings and doubly robust settings for TFP are presented in Table 49. The results when these settings were tested against Image 6, with eight replications performed, are given in Table 50. As seen in Table 50, a significant increase in performance was obtained using doubly robust settings as compared to the original robust settings. The mean value increased nearly two percent

and the LT statistic raised about four percent. Also, the variance reduced by two-thirds through the use of doubly robust settings.

Table 49. RBFNN Original and Doubly Robust Settings on TFP

Control	TFP	
	Original	Doubly Robust
DA	-2	-2
MST	6	6
BWSNR	0.1	0.1
PTSNR	6	6
BWI	0.1	0.1
TBS	0	1
CS	1	1
SIH	150	150
SIL	45	45
LSNR	14	14
WS	11	11

Table 50. RBFNN Image 6 Results on TFP

TFP - RBFNN			
	Mean	Variance	LT
Original	0.9641	0.0012	-0.9283
Doubly Robust	0.9879	0.0004	-0.9756

GRNN gradient analysis was also applied to TFP. The two combinations of settings are displayed in Table 51 and the results when replicated on Image 6 are presented in Table 52. As seen with GRNNs in TPF, doubly robust settings were unable to perform better than the original robust settings. The original robust settings already performed near perfect which made the probability of a different combination of settings to outperform the original settings nearly impossible. However, the doubly robust settings remained competitive.

Table 51. GRNN Original and Doubly Robust Settings on TFP

Control	TFP	
	Original	Doubly Robust
DA	2	-2
MST	6	6
BWSNR	0.1	0.1
PTSNR	6	1
BWI	0.1	0.1
TBS	0	0
CS	1	1
SIH	150	150
SIL	45	5
LSNR	14	14
WS	11	11

Table 52. GRNN Image 6 Results on TFP

	TFP - GRNN		
	Mean	Variance	LT
Original	0.9716	0.0009	-0.9431
Doubly Robust	0.9613	0.0007	-0.9235

The results reported in this section, as well as Section 4.5.1, validated the use of doubly robust settings to guard against system degradation. The image utilized in AutoGAD to demonstrate system degradation was not the optimal choice of the author in terms of “noisiness,” but proved useful in validating the proposed technique of determining new doubly robust settings in RPD. If new images were produced in the future with “noisier” situations, better results would be expected by using these doubly robust settings, especially with GRNNs.

4.6 AutoGAD and Factor Analysis

In Sections 4.4 and 4.5, methodology was presented on AutoGAD with each response as its own problem. This resulted in different optimal settings for each

response. However, the user was interested in optimizing all four responses with one combination of settings. Ideally, this meant detecting all targets with no false positives in as little time as possible. Possessing a fast algorithm with poor detection abilities or running a very good detection algorithm at the cost of large amounts of time was undesirable. Therefore, it was necessary to determine settings in which solving for simultaneous responses yields results that were appropriate for all four responses given any image.

To circumvent the need for subject matter experts or simple adding/subtracting of response values, factor analysis was employed to reduce the four responses into a single dimensional response. Eight different linear combination methodologies were presented in Table 17 in Section 3.5.1 which reduced the factor scores into a single dimension if multiple factors were retained.

Prior to conducting factor analysis, two of the responses were transformed to allow for simpler analysis. First, an inverse transformation was applied to time in order to make it a maximized response. Second, a (1-FPF) was applied to FPF which also made it a maximized response. Therefore, all four responses were maximized, thus making analysis of the factor scores easier to understand.

To begin factor analysis, eigenvalues were calculated which determined the amount of variation explained by different factors. Determining these eigenvalues through principal component analysis allowed for a designation of the number of factors which were retained. Three popular methods were considered when factors were retained: Kaiser Criterion, scree plots, and variance explained by factors (Dillon & Goldstein, 1984).

The Kaiser Criterion eliminated all factors that had an associated eigenvalue below one. This method can underestimate the number of factors necessary if eigenvalues of other factors score near one. Scree plots counteracted this phenomenon by examining a plot of eigenvalues versus factors. A cutoff point was selected when a noticeable drop in eigenvalue occurred, which retained some factors that scored close to one. Finally, the number of factors retained was related to the minimum amount of variance the user deemed necessary to explain. For instance, if 80 percent of the variance was necessary, factors were added until this 80 percent explanation of variance was achieved. These approaches were applied in this research to choose the appropriate number of factors and their overall results.

The eigenvalues for response data in AutoGAD is displayed in Table 53. Also displayed is the amount of variance explained by each factor (according to its eigenvalue). According to Kaiser's Criterion, only two factors were retained. This amounted to approximately 66 percent of the variance explained.

Table 53. Eigenvalues of Factors for AutoGAD

	Factor 1	Factor 2	Factor 3	Factor 4
Eigenvalue	1.5006	1.1612	0.9069	0.4314
Var. Explained	0.3752	0.2903	0.2267	0.1078

As seen in Table 53, two eigenvalues were greater than one. However, the eigenvalue related to Factor 3 was 0.9069, which was very close to one. Retaining this factor allowed for nearly 90 percent of the variance to be explained, which was suggested by a scree plot since a noticeable drop occurred between Factor 3 and Factor 4. The scree plot also coincided with the third method in which an appropriate amount of variance was

explained. Therefore, Kaiser's Criterion retained two factors and scree plots retained three factors, thus both set of factors were examined.

Following the selection of the appropriate number of factors, a factors loadings matrix was calculated. These loadings provided insight into responses that could be grouped together into a single factor. A *varimax* rotation was applied to the response data for easier interpretation of the factors. This rotation pushed the loadings further apart which allowed for a better understanding of which response belonged to which factor/component. The factors loadings matrix with two retained factors is displayed in Table 54 and the rotated factors loading matrix for two factors is given in Table 55.

Table 54. Factors Loadings Matrix for 2 Factors

	Factor 1	Factor 2
Time	0.4912	0.2192
TPF	-0.0489	0.9217
FPF	-0.82	-0.3071
TFP	-0.7646	0.4113

Table 55. Rotated Factors Loadings Matrix for 2 Factors

	Rot. Factor 1	Rot. Factor 2
Time	0.5338	0.0664
TPF	0.2221	0.8959
FPF	-0.8739	-0.0546
TFP	-0.6113	0.6164

When examining the factors loadings matrix, each response was designated to the factor in which it scored highest, regardless of the sign. For two factors in AutoGAD, Time, FPF, and TFP were combined into one factor and TPF remained its own factor. For rotated factors, Time and FPF were grouped together while TPF and TFP were combined into a single factor.

The same analysis was conducted when three factors were retained. The factors loadings matrix for three factors is given in Table 56 and the rotated matrix is presented in Table 57.

Table 56. Factors Loadings Matrix for 3 Factors

	Factor 1	Factor 2	Factor 3
Time	0.4912	0.2192	0.836
TPF	-0.0489	0.9217	-0.2507
FPF	-0.82	-0.3071	0.2402
TFP	-0.7646	0.4113	0.2955

Table 57. Rotated Factors Loadings Matrix for 3 Factors

	Rot. Factor 1	Rot. Factor 2	Rot. Factor 3
Time	0.0839	-0.013	0.9905
TPF	-0.0202	0.9561	-0.0176
FPF	-0.8168	-0.3382	-0.2071
TFP	-0.8505	0.3409	0.0378

Regardless of using factors or rotated factors, Table 56 and Table 57 both indicated that time and TPF was its own factor and that FPF and TFP was grouped into a common factor.

Following the determination of the number of factors to retain, factor (and rotated) scores were calculated. These scores represented a meaningful subspace of the original response data. Once these scores were obtained, the proposed linear combination techniques were applied to determine single dimensional response data for the AutoGAD problem. For reference, the eight linear combination techniques applied are summarized in Table 58.

Table 58. Summary of Linear Combinations on Factor Scores

Sum Factor Scores	$Y = \sum_{i=1}^n f_i$
Sum Weighted Factor Scores	$Y = \sum_{i=1}^n \lambda_i f_i$
Sum Normalized Factor Scores	$Y = \sum_{i=1}^n f_i^{\%}$
Sum Weighted Normalized Factor Scores	$Y = \sum_{i=1}^n \lambda_i f_i^{\%}$
Sum Rotated Factor Scores	$Y = \sum_{i=1}^n f_i^{(R)}$
Sum Weighted Rotated Factor Scores	$Y = \sum_{i=1}^n \lambda_i f_i^{(R)}$
Sum Normalized Rotated Factor Scores	$Y = \sum_{i=1}^n f_i^{(R^{\%})}$
Sum Weighted Normalized Rotated Factor Scores	$Y = \sum_{i=1}^n \lambda_i f_i^{(R^{\%})}$

Once the single dimensional response data was calculated, modeling approaches were conducted on the data. A quadratic regression model and an ANN was applied to the response data by utilizing the same CCD used on the responses separately, as seen in Section 4.4 and Appendix B. This resulted in eight different quadratic models and eight different ANNs for each set of factors (either two or three). After the appropriate quadratic models and well-trained ANNs were calculated, a search was performed which determined the optimal LT value and its corresponding settings.

To measure how well factor analysis performed, an alternate method of combining the response data into a single dimension was examined. This involved appropriately summing the standardized response data into a single dimension (Davis,

2009). This is the current method utilized in AutoGAD to combine the four responses into a single dimension. The final output was calculated as:

$$Combined = Time^{(st)} - TPF^{(st)} + FPF^{(st)} - TFP^{(st)} \quad (4.8)$$

Equation (4.8) was applied to the response data and a quadratic regression model was used to fit the resultant data. The LT search was applied to the model and optimal settings were calculated, as shown in Table 59.

Table 59. LT Settings for Combined Response Data

	DA	MST	BWSNR	PTSNR	BWI	TBS	CS	SIH	SIL	LSNR	WS
Combined	-2	6	0.01	6	0.03	0	0	50	5	14	3

The optimal LT settings when quadratic regression was applied to the two factor AutoGAD problem is displayed in Table 60. The optimal LT settings for ANNs on the same two factor problem are given in Table 61. As seen in the two tables, settings among the different linear combinations varied little. In addition, several techniques shared the same optimal settings. However, modeling through quadratic regression differed from modeling using ANNs in terms of optimal LT settings.

Table 60. LT Settings for Two Factors using Quadratic Regression

	DA	MST	BWSNR	PTSNR	BWI	TBS	CS	SIH	SIL	LSNR	WS
FA1	-2	14	0.01	6	0.06	0	0	50	45	14	3
FA2	-2	14	0.01	6	0.07	0	0	50	35	14	3
FA3	-2	14	0.01	5	0.06	0	0	50	45	14	3
FA4	-2	14	0.01	6	0.07	0	0	50	35	14	3
Rot-FA1	-2	14	0.01	4	0.07	0	0	50	45	14	3
Rot-FA2	-2	14	0.01	6	0.07	0	0	50	45	14	3
Rot-FA3	-2	14	0.01	5	0.07	0	0	50	45	14	3
Rot-FA4	-2	14	0.01	6	0.07	0	0	50	45	14	3

Table 61. LT Settings for Two Factors using ANNs

	DA	MST	BWSNR	PTSNR	BWI	TBS	CS	SIH	SIL	LSNR	WS
FA1	-2	6	0.1	6	0.1	1	1	50	5	14	3
FA2	-2	6	0.1	6	0.1	0	1	50	5	14	3
FA3	-2	6	0.1	6	0.01	0	1	50	45	14	3
FA4	-2	6	0.1	1	0.1	1	1	50	5	14	3
Rot-FA1	-2	6	0.1	6	0.01	1	1	150	5	14	3
Rot-FA2	-2	6	0.1	6	0.1	1	1	50	5	14	3
Rot-FA3	-2	6	0.1	6	0.01	1	1	150	5	14	3
Rot-FA4	-2	6	0.1	1	0.1	0	1	50	5	14	3

To measure the performance of these settings versus the combined settings calculated in Equation (4.8), the settings were applied to the untrained images, as done in Section 4.4. Each combination of settings was applied to each of the eight images. The results under each of the four responses were averaged, variance calculated, and an LT statistic was also calculated.

As seen in Table 62, all factor analysis methods outperformed the combined setting results in terms of FPF and TFP. Also, most of the methods were superior in terms of time. Only two of the rotated factor methods were better than the combined settings in terms of TPF. From these results, a strong case can be made that using factor analysis techniques to reduce the data set was preferred over simply summing the standardized response data. No one factor (rotated) method clearly outperformed another, but rather all outperformed the combined settings three out of the four available responses almost every time.

Table 62. LT Results for Two Factors

	Method	Time	TPF	FPF	TFP
	Combined	1456.387	-0.83286	0.002415	-0.01681
Quad. Reg.	FA1	1334.624	-0.57549	1.58E-05	-0.48553
	FA2	1555.191	-0.55978	9.41E-06	-0.54581
	FA3	1551.976	-0.57082	2.2E-05	-0.45629
	FA4	1465.315	-0.56009	1.02E-05	-0.54023
	Rot-FA1	1400.619	-0.55851	1.05E-05	-0.52534
	Rot-FA2	1555.191	-0.55978	9.41E-06	-0.54581
	Rot-FA3	1447.621	-0.56272	1.17E-05	-0.51857
	Rot-FA4	1555.191	-0.55978	9.41E-06	-0.54581
ANNs	FA1	1474.466	-0.69037	2.66E-05	-0.52894
	FA2	1522.444	-0.67242	1.49E-05	-0.63267
	FA3	1721.981	-0.81845	0.000914	-0.29161
	FA4	1441.736	-0.69715	4.25E-05	-0.49343
	Rot-FA1	1779.229	-0.84474	0.002124	-0.09883
	Rot-FA2	1474.466	-0.69037	2.66E-05	-0.52894
	Rot-FA3	1779.229	-0.84474	0.002124	-0.09883
	Rot-FA4	1350.419	-0.69106	3.09E-05	-0.56769

The same analysis was conducted using three retained factors to see if any performance was gained by adding another factor. The optimal LT settings using quadratic regression when three factors were retained are given in Table 63. Also, the LT settings for the same three factors using ANNs are presented in Table 64. Once again, little variation among the settings within each approach occurred, although there was more variation compared to settings in Table 60 and Table 61.

Table 63. LT Settings for Three Factors using Quadratic Regression

	DA	MST	BWSNR	PTSNR	BWI	TBS	CS	SIH	SIL	LSNR	WS
FA1	-2	8	0.01	6	0.1	0	0	50	5	14	3
FA2	-2	9	0.01	6	0.1	0	0	50	5	14	3
FA3	-2	13	0.01	6	0.1	0	0	50	45	14	3
FA4	-2	14	0.1	1	0.1	0	1	150	5	4	3
Rot-FA1	-2	14	0.01	6	0.07	0	0	50	45	14	3
Rot-FA2	-2	14	0.01	6	0.07	0	0	50	35	14	3
Rot-FA3	-2	14	0.01	4	0.07	0	0	50	45	14	3
Rot-FA4	-2	10	0.1	6	0.1	0	0	150	5	14	3

Table 64. LT Settings for Three Factors using ANNs

	DA	MST	BWSNR	PTSNR	BWI	TBS	CS	SIH	SIL	LSNR	WS
FA1	-2	6	0.1	6	0.01	0	1	50	5	14	3
FA2	-2	6	0.1	6	0.1	0	1	50	5	14	3
FA3	-2	6	0.1	6	0.1	1	1	50	5	14	3
FA4	-2	6	0.1	6	0.1	1	1	50	5	14	3
Rot-FA1	-2	6	0.1	6	0.01	0	1	150	45	4	3
Rot-FA2	-2	6	0.1	6	0.1	0	1	50	5	14	3
Rot-FA3	-2	6	0.1	6	0.01	0	1	50	5	14	3
Rot-FA4	-2	6	0.1	6	0.1	1	1	50	5	14	3

These settings were applied to the eight untrained images and their appropriate statistics were collected. As seen in Table 65, stronger results were obtained when three factors were retained, as compared to two. Once again, all methods outperformed the combined settings in terms of FPF and TFP. In fact, the margin between the two was calculated as much larger with comparison to two factors retained. TPF for the combined settings remained superior, but the gap was reduced when using three factors. Finally, half of the factor analysis methods outperformed the combined settings in terms of time,

while the other half remained competitive. Overall, the factor (rotated) methods were preferred over the combined settings, regardless of the approach utilized, due to the substantial reduction of LT values for FPF and TFP, while remaining better or competitive in terms of time and TPF.

Table 65. LT Results for Three Factors

	Method	Time	TPF	FPF	TFP
	Combined	1456.387	-0.83286	0.002415	-0.01681
Quad. Reg.	FA1	1470.449	-0.61335	3.36E-05	-0.46849
	FA2	1444.256	-0.61528	3.33E-05	-0.45931
	FA3	1561.006	-0.61244	2.56E-05	-0.52579
	FA4	1540.971	-0.57881	2.11E-06	-0.73797
	Rot-FA1	1555.191	-0.55978	9.41E-06	-0.54581
	Rot-FA2	1465.315	-0.56009	1.02E-05	-0.54023
	Rot-FA3	1400.619	-0.55851	1.05E-05	-0.52534
	Rot-FA4	1528.943	-0.62177	2.84E-05	-0.53144
ANNs	FA1	1532.336	-0.84592	0.001098	-0.15453
	FA2	1522.444	-0.67242	1.49E-05	-0.63267
	FA3	1474.466	-0.69037	2.66E-05	-0.52894
	FA4	1474.466	-0.69037	2.66E-05	-0.52894
	Rot-FA1	1602.298	-0.82344	0.00084	-0.30052
	Rot-FA2	1522.444	-0.67242	1.49E-05	-0.63267
	Rot-FA3	1721.981	-0.81845	0.000914	-0.29161
	Rot-FA4	1474.466	-0.69037	2.66E-05	-0.52894

The results reported in this section using factor analysis were strong in terms of their use over simply summing standardized response data. Although superior performance was not achieved in terms of all four outputs simultaneously, factor analysis typically had improved results in three of the four outputs. Also, the difference in LT scores for FPF and TFP was quite large when using factor analysis over simple summation. Also, ANNs provided a better fit to the data over quadratic regression, but this situation could change depending on the nature of the problem being examined.

V. Contributions and Future Research

5.1 Overview

A summary of the contributions made to the field of applied statistics and design of experiments through the research conducted and presented in this document is provided in this chapter. A list of potential areas for further investigation related to this research is also provided.

5.2 Research Contributions

Several contributions in the fields of applied statistics and design of experiments were made in this research. Each contribution is summarized.

5.2.1 Doubly Robust Settings

A gradient analysis was applied to the coefficients of derived process models. This gradient search determined the worst possible system degradation that could occur, through perturbations in the coefficients. Solving for robust control settings along this gradient search allowed for future protection against unfavorable results due to degradation. These doubly robust settings maintained their consistency in being robust to noise in the system, as modeled in traditional RPD. A gradient search was developed using quadratic regression, RBFNNs or GRNNs. This gradient analysis is applicable to any process model involving control and noise variables.

5.2.2 Artificial Neural Networks in RPD

Methods for utilizing ANNs in RPD were discussed in this dissertation. Two methods, depending on available data, were derived to model the mean and variance models necessary for RPD. Experiments confirmed the usefulness of ANNs when quadratic regression failed to fit highly non-linear models.

A gradient search algorithm was developed based upon the weights of the ANNs to determine doubly robust settings if quadratic regression is inappropriate. The doubly robust settings in ANNs proved as effective, if not more so, than those obtained using quadratic regression.

5.2.3 Factor Analysis in RPD

An alternative set of methods was derived to reduce multiple response problems to a single dimension. Ideally, factor analysis would retain only one factor; however, if multiple factors still remained, linear combinations were applied to reduce the application to a single response. Reduction to a single response allowed for RPD to be performed in the traditional sense. Factor analysis was shown to be more effective than simply a summation of standardized response data.

5.3 Recommendations for Future Research

Several areas of continued research are suggested.

5.3.1 Robust Parameter Design

Throughout this research, Lin & Tu (1998) methodology was applied to RPD and doubly robust RPD. Table 1 presented alternative methods of solving the dual response

problem in RPD and LT was selected as the author's choice. Other methods could have provided more appropriate results depending on the problem. The difference between utilizing the different methods is left to be explored.

5.2.2 Artificial Neural Networks

RBFNNs and GRNNs were selected as the neural networks applied in this research. However, the Feed Forward neural network (FFNN) is another widely used ANN in research. Typically used for classification purposes, this ANN can be tested for its validity in RPD. Also, a gradient analysis could be conducted on the weights of the FFNN to determine doubly robust settings.

5.2.3 Multiple Responses in RPD

Eight linear combinations of factor analysis were developed to reduce multiple responses into a single dimension. Further exploration of this concept could discover a new combination technique which could achieve superior and more consistent results. Multiple responses in RPD is a lightly researched area and has the potential for tremendous contributions.

Appendix A. Summary of Example Problems

Problem	Source	Variables	Responses	Application
Semiconductor Manufacturing	Myers & Montgomery (2002:566)	2 control 3 noise	Single (minimized)	Quadratic Regression in RPD Doubly Robust (QR)
Semiconductor Manufacturing Extended (higher-order terms)	Myers & Montgomery (2002:566)	2 control 3 noise	Single (minimized)	ANNs in RPD Doubly Robust (ANNs)
Force Transducer	Koksoy (2008)	3 control 2 noise	Non-linearity (u=1) Hysteresis (min)	QR vs ANNs in RPD
Notional $3^2 \times 2^2$ Design with 5 Responses	None	2 control 2 noise	Y_1, Y_2, Y_3 (min) Y_4, Y_5 (max)	Factor Analysis

Problem (examined but not presented)	Source	Variables	Responses	Application
Color TV Images	Myers & Montgomery (2002:570)	2 control 2 noise	Single (maximized)	Quadratic Regression and ANNs in RPD Doubly Robust (QR)
Hard Disk Drive Quality	Su & Tong (1997)	5 control 1 noise	PW (min) HFA (max) OW (max) PS (min)	QR vs ANNs in RPD
Notional $3^2 \times 2^2$ Design with 4 Responses (representative of Davis (2009) work)	None	2 control 2 noise	Y_1, Y_2 (max) Y_3, Y_4 (min)	Factor Analysis

Appendix B. CCD for AutoGAD

Dim	Max	Bin	Pt SNR	Bin ID	Thresh	Clean	High	Low	LSNR	Win	Fishers	% Tgt	# Clusters
-2	6	0.1	6	0.1	1	0	150	5	14	3	-1.00	-0.80	-0.26
-2	6	0.01	1	0.01	0	1	150	45	14	11	0.71	-0.63	-1.00
-2	14	0.1	6	0.1	1	0	50	5	4	3	1.00	-0.19	-0.99
-2	14	0.1	6	0.1	1	0	150	5	4	11	0.71	-1.00	0.26
-2	14	0.1	1	0.01	0	0	150	45	14	3	1.00	-1.00	1.00
-2	14	0.1	6	0.1	0	1	150	45	4	3	0.71	-0.63	-1.00
2	10	0.06	3.5	0.06	1	0	100	25	9	7	-1.00	1.00	-0.79
2	6	0.1	1	0.01	1	0	150	45	14	11	0.43	-0.84	-0.56
2	14	0.01	1	0.1	0	1	50	45	4	3	1.00	-1.00	1.00
2	14	0.1	6	0.01	0	1	50	5	14	3	0.71	-1.00	0.26
2	14	0.01	6	0.1	0	0	50	5	14	11	0.71	-0.63	-1.00
-2	6	0.01	6	0.01	0	0	150	5	4	3	-1.00	-0.80	-0.26
-2	6	0.01	6	0.1	1	0	50	45	4	3	0.43	-0.84	-0.56
2	14	0.01	6	0.01	0	0	50	45	14	3	0.71	-1.00	0.26
2	14	0.01	6	0.1	1	1	50	45	4	11	-1.00	1.00	-0.79
2	14	0.1	6	0.01	0	1	50	5	4	11	0.71	-0.63	-1.00
-2	6	0.1	1	0.1	0	0	50	45	14	3	0.71	-1.00	0.24
2	6	0.1	1	0.01	1	0	150	45	14	11	1.00	-0.19	-0.99
-2	6	0.1	6	0.01	1	0	50	5	14	3	-1.00	-0.80	-0.26
-2	14	0.01	1	0.01	1	0	50	45	4	3	0.71	-1.00	0.26
2	6	0.01	6	0.1	1	0	50	5	4	3	0.71	-1.00	0.24
2	14	0.1	1	0.1	0	1	50	5	4	11	-1.00	-0.80	-0.26
-2	14	0.1	6	0.01	1	1	50	45	14	11	0.43	-0.84	-0.56
2	14	0.01	6	0.1	1	0	150	45	4	11	0.71	-0.63	-1.00
2	14	0.01	6	0.01	0	0	50	5	14	11	0.71	-1.00	0.24
2	6	0.01	6	0.01	1	0	50	45	14	3	1.00	-0.19	-0.99
-2	14	0.01	6	0.01	1	0	150	45	4	3	0.43	-0.84	-0.56
2	6	0.1	6	0.1	0	0	150	5	4	11	-1.00	1.00	-0.79
-2	6	0.1	1	0.01	0	1	150	5	4	11	1.00	-1.00	1.00
-2	6	0.1	1	0.1	0	0	150	45	4	11	0.71	-1.00	0.26
2	6	0.1	6	0.1	1	1	150	5	4	11	-1.00	-0.80	-0.26
2	6	0.1	6	0.01	1	1	150	45	4	11	0.43	-0.84	-0.56
-2	14	0.01	6	0.01	0	1	150	45	4	3	1.00	-1.00	1.00
-2	6	0.1	6	0.01	0	0	150	45	14	3	0.71	-0.63	-1.00
2	6	0.01	6	0.1	1	0	50	5	4	3	-1.00	-0.80	-0.26
2	14	0.01	6	0.01	0	0	150	45	4	3	0.71	-1.00	0.26
2	14	0.1	6	0.01	1	1	150	5	4	3	1.00	-1.00	1.00
2	14	0.1	1	0.01	1	0	50	5	14	3	-1.00	-0.80	-0.26
-2	14	0.01	6	0.01	0	1	150	5	4	3	1.00	-1.00	1.00
-2	6	0.1	1	0.01	1	1	150	45	14	3	0.71	-0.63	-1.00
-2	14	0.1	6	0.1	0	1	50	5	4	3	-1.00	1.00	-0.79
-2	6	0.1	6	0.1	0	1	150	5	14	3	0.71	-1.00	0.24
0	10	0.06	3.5	0.06	0	1	100	5	9	7	0.71	-1.00	0.24
2	14	0.01	6	0.1	0	0	50	5	14	3	0.43	-0.84	-0.56
-2	6	0.1	1	0.1	0	1	150	5	4	11	-1.00	-0.80	-0.26
-2	6	0.01	6	0.01	1	0	50	5	4	3	0.71	-0.63	-1.00
-2	6	0.1	1	0.1	1	1	150	45	4	3	-1.00	1.00	-0.79
2	6	0.01	1	0.1	0	1	50	45	4	11	0.71	-1.00	0.24
0	10	0.06	3.5	0.1	1	0	100	25	9	7	0.71	-1.00	0.26

Bibliography

- Anderson, T.W. (2003). *An Introduction to Multivariate Statistical Analysis (3rd ed)*. New York: Wiley.
- Box, G.E.P. & Wilson, K.B. (1951). "On the experimental attainment of optimum conditions." *Journal of the Royal Statistical Society, Series B*, 13: 1-45.
- Brenneman, W.A. & Myers, W.R. (2003). "Robust parameter design with categorical noise variables." *Journal of Quality Technology*, 35: 335-341.
- Carnot, S., Clapeyron, E., & Clausius, R. (2005). *Reflections on the Motive Power of Fire – and other Papers on the Second Law of Thermodynamics*. New York: Dover Publications.
- Chang, H.H. (2006). "Dynamic multi-response experiments by backpropagation networks and desirability functions." *Journal of the Chinese Institute of Industrial Engineers*, 23 (4): 280-288.
- Chang, H.H. (2008). "A data mining approach to dynamic multiple responses in Taguchi experimental design." *Expert Systems with Applications*, 35: 1095-1103.
- Copeland, K.A.F. & Nelson, P.R. (1996). "Dual response optimization via direct function minimization." *Journal of Quality Technology*, 28: 331-336.
- Davis, M. (2009). *Using Multiple Robust Parameter Design Techniques to Improve Hyperspectral Anomaly Detection Algorithm Performance*. Thesis, Air Force Institute of Technology, WPAFB, OH.
- Del Castillo, E. & Montgomery, D.C. (1993). "A nonlinear programming solution to the dual response problem." *Journal of Quality Technology*, 25: 199-204.
- Derringer, G. & Suich, R. (1980). "Simultaneous optimization of several response variables." *Journal of Quality Technology*, 12: 214-219.
- Dillon, W.R. & Goldstein, M. (1984). *Multivariate Analysis: Methods and Applications*. New York: Wiley.
- Duda, R.O., Hart, P.E., & Stork, D.G. (2001). *Pattern Classification (2nd ed.)*. New York, NY: John Wiley & Sons.
- Harrington, E.C. (1965). "The desirability function." *Industrial Quality Control*, 21: 494-498.

- Johnson, R.J. *Improved Feature Extraction, Feature Selection, and Identification Techniques that Create a Fast Unsupervised Hyperspectral Target Detection Algorithm*. MS thesis, Air Force Institute of Technology, WPAFB OH, 2008.
- Kirkwood, C.W. (1997). *Strategic Decision Making*. Belmont, CA: Duxbury Press.
- Koksoy, O. (2008). "A nonlinear programming solution to robust multi-response quality problem." *Applied Mathematics and Computation*, 196: 603-612.
- Kuhnt, S. & Erdbrugge, M. (2004). "A strategy of robust parameter design for multiple responses." *Statistical Modeling*, 4: 249-264.
- Kuncheva, L.L. (2004). *Combining Pattern Classifiers: Methods and Algorithms*. New Jersey: John Wiley & Sons, Inc.
- Landgrebe, D.A. *Signal Theory Methods in Multispectral Remote Sensing*. John Wiley and Sons, Inc., 2003.
- Lin, D.K.J. & Tu, W. (1995). "Dual response surface optimization." *Journal of Quality Technology*, 27: 34-39.
- Lind, E.E., Goldin, J., & Hickman, J.B. (1960). "Fitting yield and cost response surfaces." *Chemical Engineering Progress*, 56: 62-68.
- Loeffelholz, B.J. & Bauer, K.W. (2009). "An Examination into Robust Parameter Design," *Proceedings of the Huntsville Simulation Conference*, Huntsville, AL, Oct 27-29, 2009.
- Loeffelholz, B.J., Bednar, E., & Bauer, K.W. (2009). "Predicting NBA games using neural networks," *Journal of Quantitative Analysis in Sports*, 5 (1), Article 7.
- Lohninger, H. *Teach/Me Data Analysis*. Berlin: Springer-Verlag, 1999.
- Looney, C.G. (1997). *Pattern Recognition Using Neural Networks*. New York: Oxford University Press.
- Miller, M.K. *Exploitation of Intra-Spectral Band Correlation for Rapid Feature Selection, and Target Identification in Hyperspectral Imagery*. MS thesis, Air Force Institute of Technology, WPAFB OH, 2009.
- Mindrup, F.M., Bihl, T.J., & Bauer, K.W. (2010). "Modeling noise in a framework to optimize the detection of anomalies in hyperspectral imaging," *Intelligent Engineering Systems through Artificial Neural Networks*, Submitted 1 June, 2010.

- Montgomery, D.C., Peck, E.A., & Vining, G.G. (2004). *Introduction to Linear Regression Analysis* (3rd ed). Wiley: New York.
- Myers, R.H. & Carter, W.H. (1973). "Response surface techniques for dual response systems." *Technometrics*, 15: 301-317.
- Myers, R.H. & Montgomery, D.C. (2002). *Response Surface Methodology: Process and Product Optimization Using Design Experiments* (2nd ed). Wiley: New York.
- Myers, R.H., Montgomery, D.C., & Vining, G.G. (2002). *Generalized Linear Models: With Applications in Engineering and the Sciences*. Wiley: New York.
- Pabich, P.J. "Hyperspectral imagery: Warfighting through a different set of eyes." *Occasional Paper 31*, Center for Strategy and Technology, Air War College, Air University, 2002.
- Robinson, T.J., Borror, C.M., & Myers, R.H. (2004). "Robust parameter design: A review." *Quality and Reliability Engineering International*, 20: 81-101.
- Sanchez, S.M. (1994). "A robust design tutorial." *Proceedings of the 1994 Winter Simulation Conference*, Institute of Electrical and Electronic Engineers, Piscataway, NJ, 106-113.
- Shaibu, A.B. & Cho, B.R. (2009). "Another view of dual response surface modeling and optimization in robust parameter design." *International Journal of Advanced Manufacturing and Technology*, 41: 631-641.
- Shaw, G. & Manolakis, D. "Signal processing for Hyperspectral image exploitation." *IEEE Signal Processing Magazine*, 19 (1), 2002: 12-16.
- Smetek, T.E. *Hyperspectral Imagery Target Detection using Improved Anomaly Detection and Signature Matching Methods*. Dissertation, Air Force Institute of Technology, WPAFB OH, 2007, AFIT/DS/ENS/07.
- Specht, D.F. (1991). "A general regression neural network." *IEEE Transactions on Neural Networks*, 2 (6): 568-576.
- Su, C.T. & Tong, L.I. (1997). "Multi-response robust design by principal component analysis." *Total Quality Management*, 8 (6): 409-416.
- Taguchi, G. (1986). *Introduction to Quality Engineering*. White Plains, NY: UNIPUB/Kraus International.
- Tang, L.C. & Xu, K. (2002). "A unified approach for dual response surface optimization." *Journal of Quality Technology*, 34 (4): 437-447.

- Thurstone, L.L. (1947). *Multiple Factor Analysis*. Chicago: University of Chicago Press.
- Vining, G.G. & Myers, R.H. (1990). "Combining Taguchi and response surface philosophies: A dual response approach." *Journal of Quality Technology*, 22: 38-45.
- Wasserman, P.D. (1993). *Advanced Methods in Neural Computing*. New York: Van Nostrand Reinhold.
- Williams, J.P. *Robustness of Multiple Clustering Algorithms on Hyperspectral Images*. MS thesis, Air Force Institute of Technology, WPAFB OH, 2007.
- Zandieh, M., Amiri, M., Vahdani, B., & Soltani, R. (2009). "A robust parameter design for multi-response problems." *Journal of Computational and Applied Mathematics*, 230: 463-476.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
<p>The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.</p> <p>PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.</p>					
1. REPORT DATE (DD-MM-YYYY) 17-09-2010		2. REPORT TYPE Ph.D Dissertation		3. DATES COVERED (From – To) Oct 2006 – Sep 2010	
4. TITLE AND SUBTITLE Neural Extensions to Robust Parameter Design				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Loeffelholz, Bernard Jacob				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Street, Building 642 WPAFB OH 45433-7765				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT/DS/ENS/10-03	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) John Jacobson 4180 Watson Way WPAFB OH 45433-7765				10. SPONSOR/MONITOR'S ACRONYM(S) NASIC/DAIA AFRL/RYZT	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT <p>Robust parameter design (RPD) is implemented in systems in which a user wants to minimize the variance of a system response caused by uncontrollable factors while obtaining a consistent and reliable system response over time. We propose the use of artificial neural networks to compensate for highly non-linear problems that quadratic regression fails to accurately model. RPD is conducted under the assumption that the relationship between system response and controllable and uncontrollable variables does not change over time. We propose a methodology to find a new set of settings that will be robust to moderate system degradation while remaining robust to noise variables within the system RPD has been well developed on single response problems. Sparse literature exists on dealing with multiple responses in RPD and most methods utilize a subjective weighting scheme. To account for multiple responses, we examine the use of factor analysis on the response data. All the proposed techniques are applied to textbook applications to demonstrate their utility. An Air Force application problem is examined to demonstrate the new technique's potential on a real-world problem that is highly non-linear. The application is a detector developed to detect anomalies within hyper-spectral imagery.</p>					
15. SUBJECT TERMS Robust Parameter Design, Artificial Neural Networks, System Degradation, Factor Analysis, Multiple Responses					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPOR T	b. ABSTRAC T	c. THIS PAGE			Dr. Kenneth W. Bauer (ENS)
U	U	U	UU	168	19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4328; e-mail: Kenneth.Bauer@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18